Personal Computer

NEC PC-8801

プログラミング入門

野々山隆幸＝著

ナツメ社

Personal Computer
PC-8801
NEC
プログラミング入門
野々山隆幸＝著
ナツメ社

[本書を読まれる方へ]

⦿いま、PC-8801を前にして、
ちょっとフレッシュな、そして、
メカアレルギーからくる不安とがまじりあった、
そんな心もちではありませんか？
⦿パソコンにはいろいろなことができるといわれています。
確かに、そのとおりなのですが、実をいうと、
考えていたほどたやすくはいかないのです。
こんなことをいうと、少々がっかりするかも
しれませんが、まあ、気を落ちつけてください。
⦿どんなことでも、効率を上げさらに
向上しようと思えば、困難はつきものです。
このことはパソコンにもあてはまるのです。
⦿そのかわり、努力した結果、パソコンを理解し、
自分の仕事にどう組み入れていくかがわかったとき、
あなたの、ビジネスでの思考様式は
ずいぶん変わってくるはずです。
⦿そのための手段として、まず、あなたが
パソコンに何をやらせたいかを念頭において、
ベーシック・プログラミングを学んでいくことが必要です。
⦿この本では、使い方の実際から具体的な
プログラミングまで、書いてあるとおりに
やればよいように編集してあります。
⦿さあ、PC-8801に取り組んでみよう！

cover design——nobuo nakagaki+yoshifumi hayase

Personal Computer

PC NEC 8801

プログラミング入門

野々山隆幸一著

ナツメ社

はじめに

ビジネス志向のＰＣ-８８０１

ＮＥＣパーソナルコンピュータＰＣ-８８０１が脚光をあびています。大容量のメモリには、強力なN_{88}－ＢＡＳＩＣを搭載しています。ベストセラーＰＣ-８００１用プログラムもそのまま使えます。周辺機器もひとクラス上のものばかりです。ＰＣ-８８０１は、待望のビジネス志向のパソコンです。

本書の構成

本書は、はじめてパソコンにさわるという人を対象にして、ＰＣ-８８０１のプログラミングについて、わかりやすく説明しています。本書の前半では、電源ＯＮ、キー操作、デモストレーション、電卓がわりに使う方法など、ＰＣ-８８０１の基本的な操作や機能について述べています。また、後半では、経営分析、設備投資計画、利益計画、予算実績差異分析などビジネス志向のプログラムについて説明し、メニュー画面、棒グラフ、損益分岐点図表、利益計画用損益計算書、漢字見出しによる予算実績差異分析表など図表についても述べています。

習うより慣れろ

しかし、本書をたんに読むだけでは、あまり効果はありません。だいじなことは、本書を読みながら、実際にＰＣ-８８０１にさわって動かすということです。よほど無茶なことをしないかぎり、どんな操作をしてもＰＣ-８８０１はこわれることはありません。どんどんキーをたたいてください。習うより慣れろです。

なお、本書の執筆にあたって、ＮＥＣ日本電気株式会社のパーソナルコンピュータ事業部および販売部のなみなみならぬご協力をいただきました。深く感謝いたします。また、ナツメ社編集部の斎藤洋氏には、本書の制作面でたいへんお世話になりました。心からお礼を申し上げます。そして、夜遅くまでがんばってくれたＰＣ-８８０１にも感謝しています。

ＰＣ-８８０１を動かすのは「あなた」

それでは、本書を持ってＰＣ-８８０１の前にすわってください。ＰＣ-８８０１を動かすのは「あなた」です。

野々山隆幸

もくじ

第0章

準備はいいですか

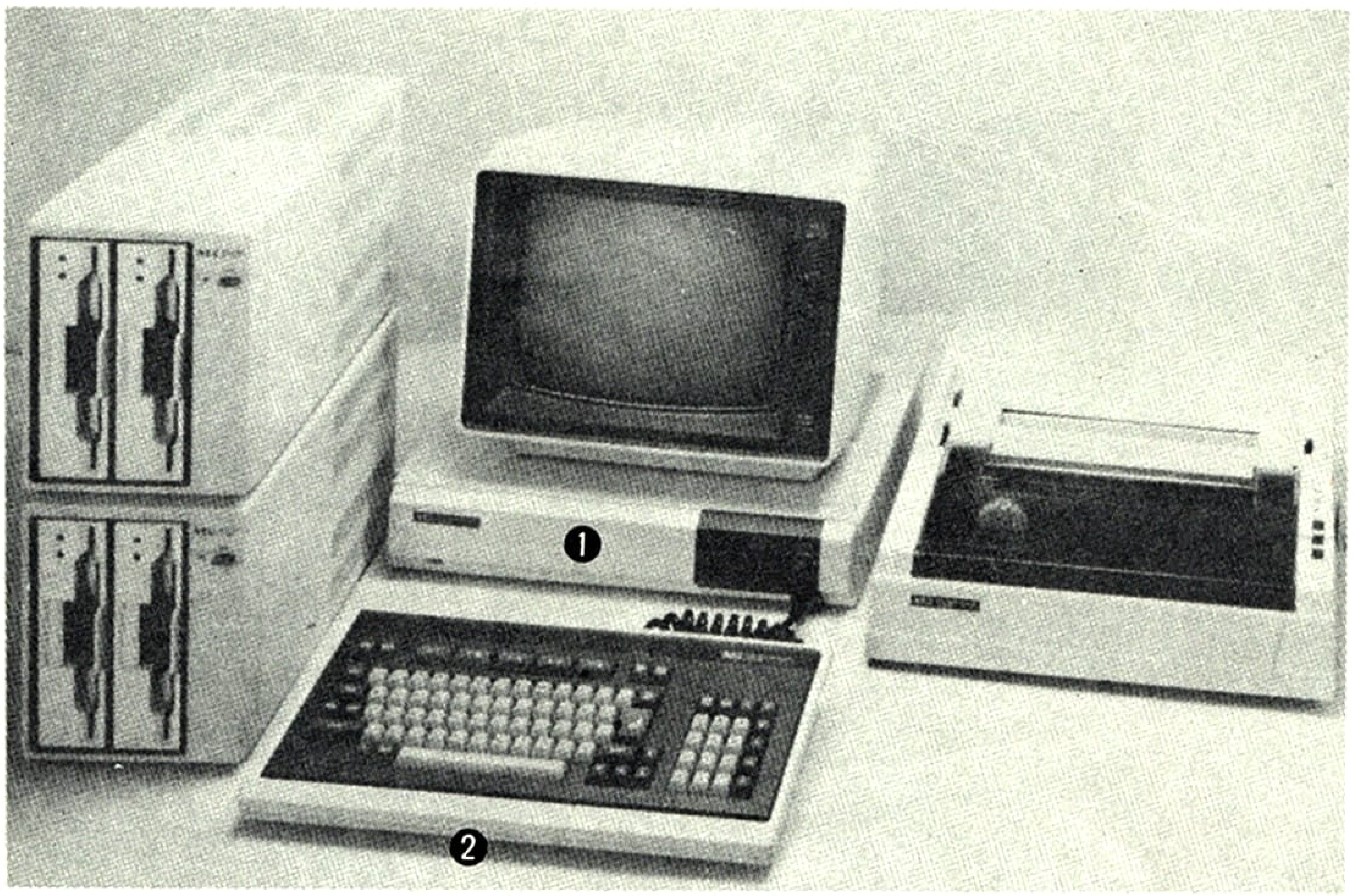
0

0・1 PC-8801の前にすわろう

ＰＣ-８８０１とは？

PC-8801

ＰＣ-８８０１の前にすわってください。
あなたの目の前には、何がありますか。
テレビとタイプライタと……。そのほかにも、何か装置がありますね。
これら全部がＰＣ-８８０１でしょうか。ちがいます。
ＰＣ-８８０１とは、テレビのようなものの下にある装置とタイプライタのようなものの２つをいいます。



❶ 本　　体 ｝ＰＣ-８８０１
❷ キーボード ｝

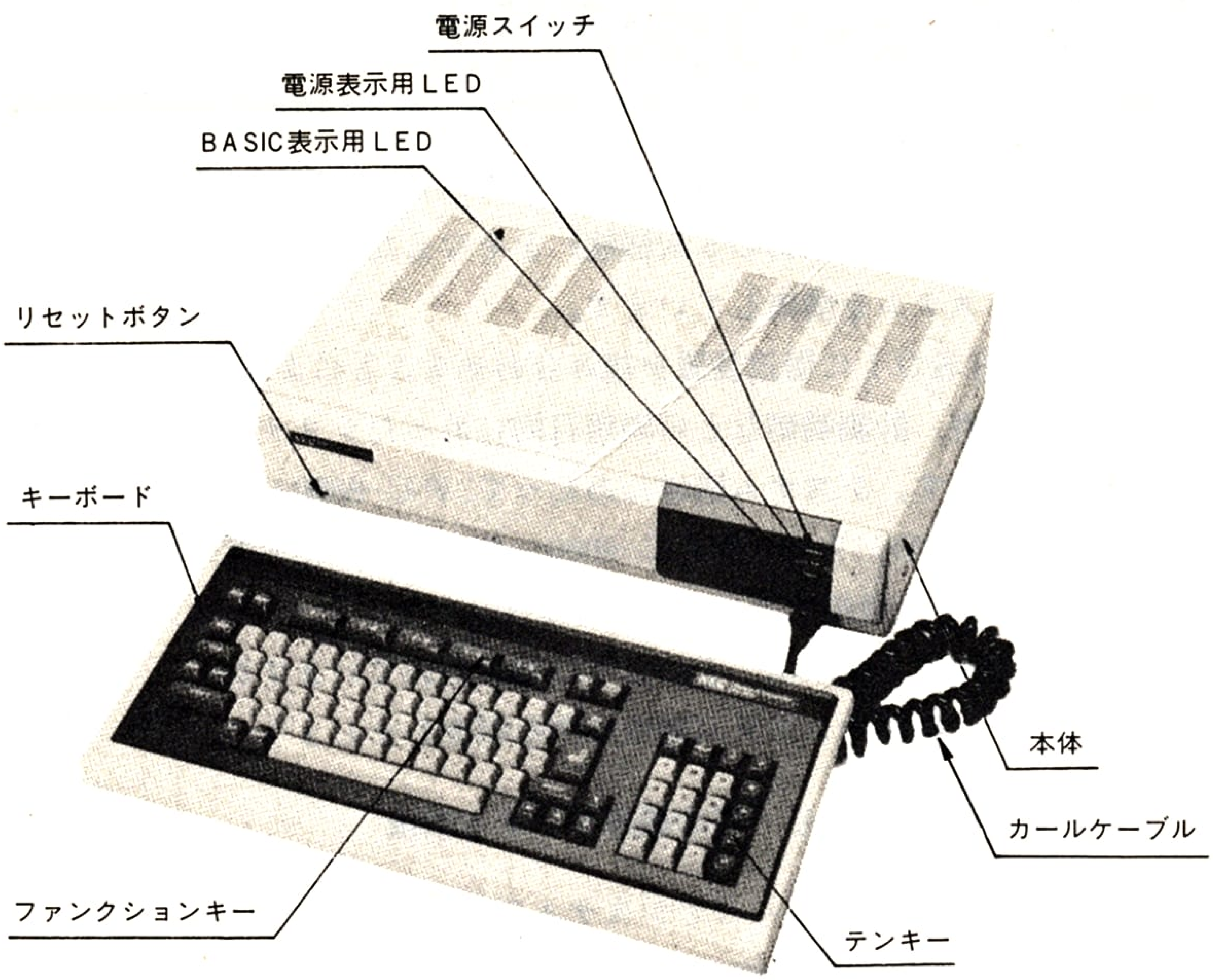
ＰＣ-８８０１の外観は？

ＰＣ-８８０１を正面からみてみましょう。

本体
キーボード

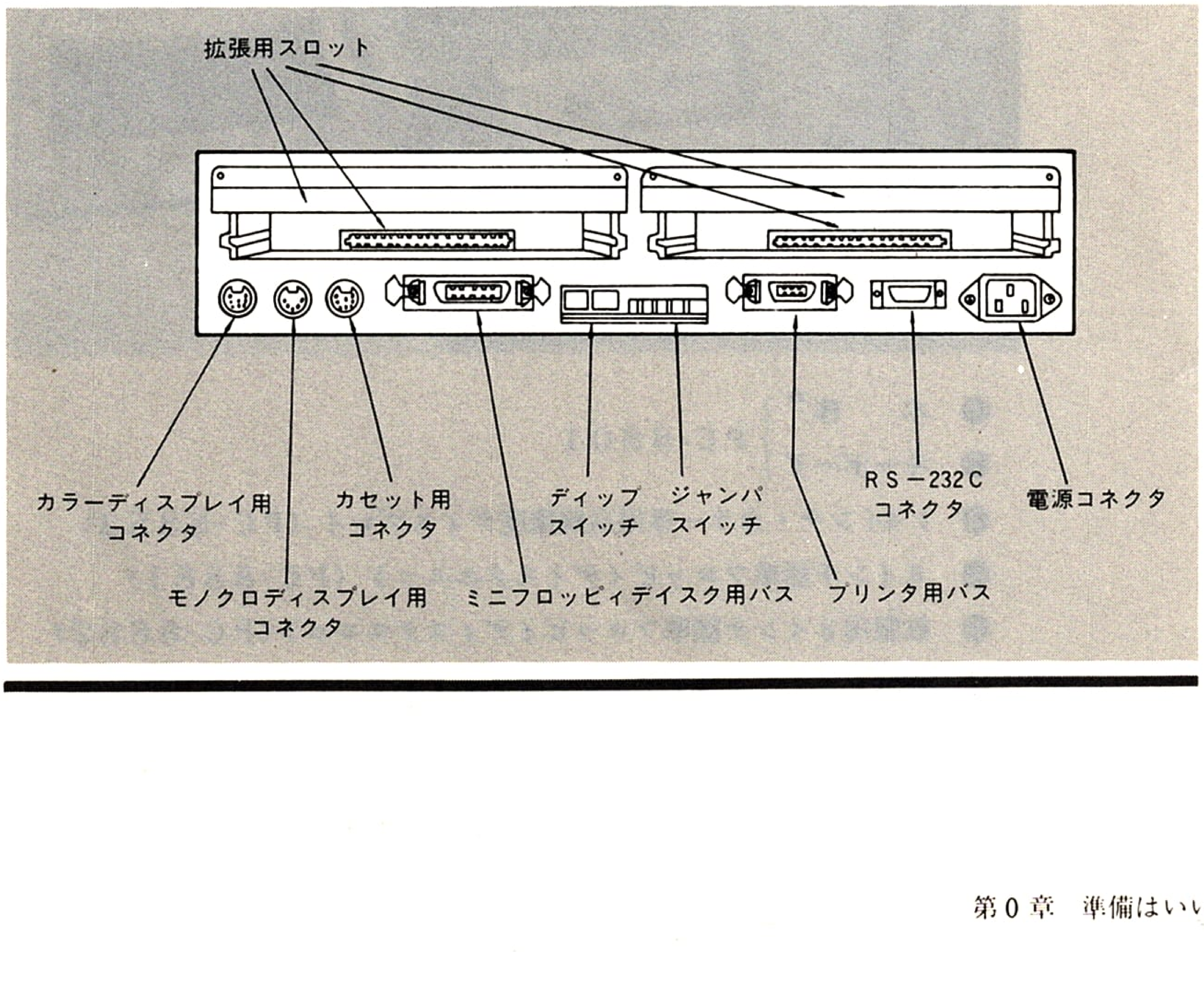
ＰＣ-８８０１は、本体とキーボードに分かれています。テレビのようなものの下にある装置が、本体です。タイプライタのようなものをキーボードといいます。そして、本体とキーボードは、カールケーブルで接続されています。ここでは、とりあえず、本体の前面右にある電源スイッチおよび前面左にあるリセットボタンの位置だけを確認しておきましょう。

電源スイッチ
リセットボタン



次に、ＰＣ-８８０１本体の背面を見てみましょう。

ケーブルによって、ＰＣ-８８０１がいろいろな装置と接続されていますね。



0・2　システムの構成は

　パーソナルコンピュータは、コンピュータの基本的機能（入力、出力、記憶、演算、制御の５大機能）をそなえた装置が組み合わされて、１つのシステムを構成しています。

システム構成

　ＰＣ−８８０１に周辺装置を組み合わせることによって、事務処理、端末、教育、計測制御などの利用目的にあったシステムを構成することができます。システム構成にあたって、ＰＣ-8800シリーズだけでなく、ＰＣ-8000シリーズやＰＣ-6000シリーズの周辺装置の多くは、ＰＣ−８８０１に接続することができます。

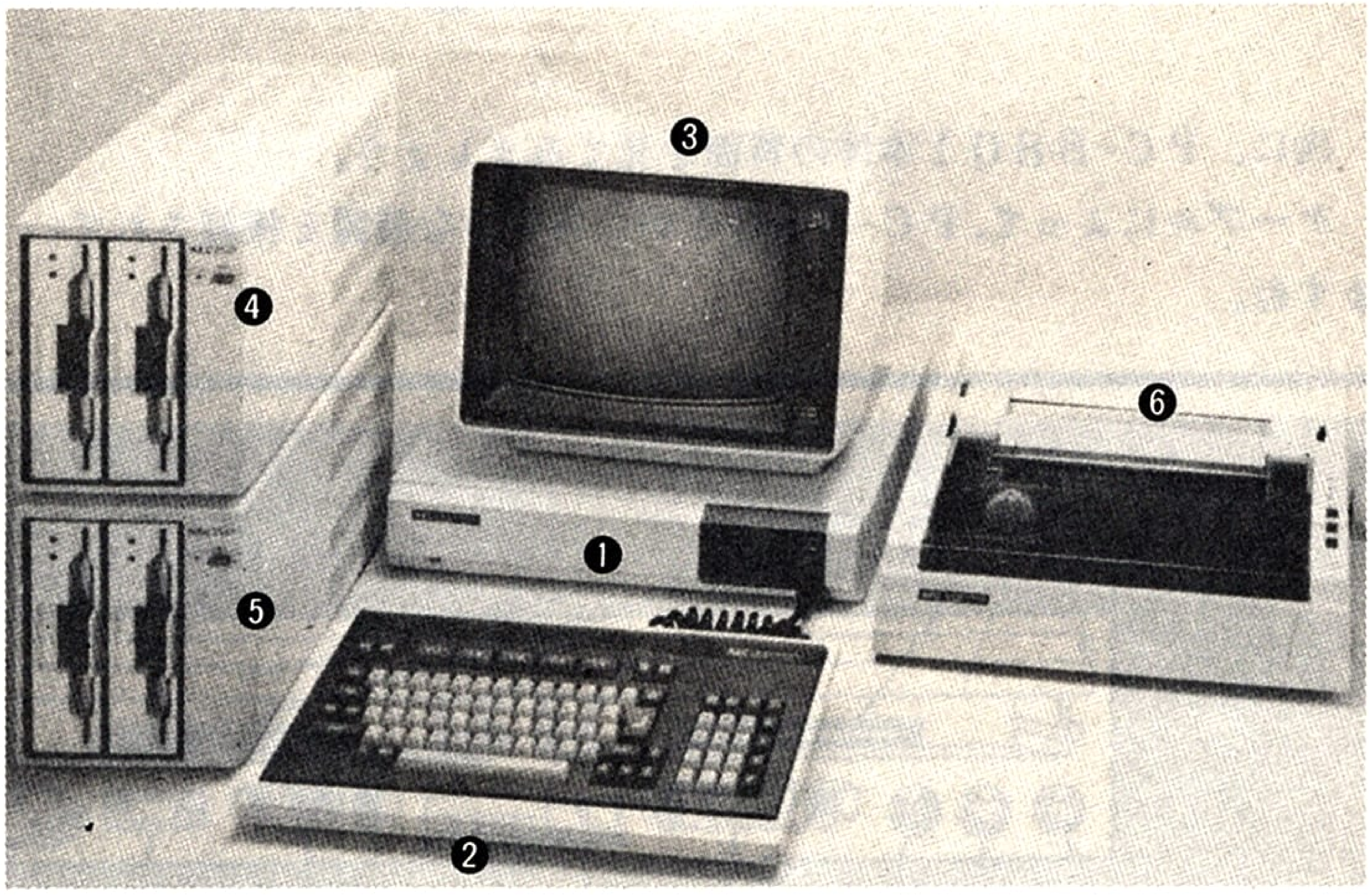
　あなたのシステムは、どのように構成されていますか。

　次に、３種類のシステム構成の例を紹介しましょう。

ＰＣ−８８００シリーズの標準システム構成

ＰＣ−８８００シリーズ

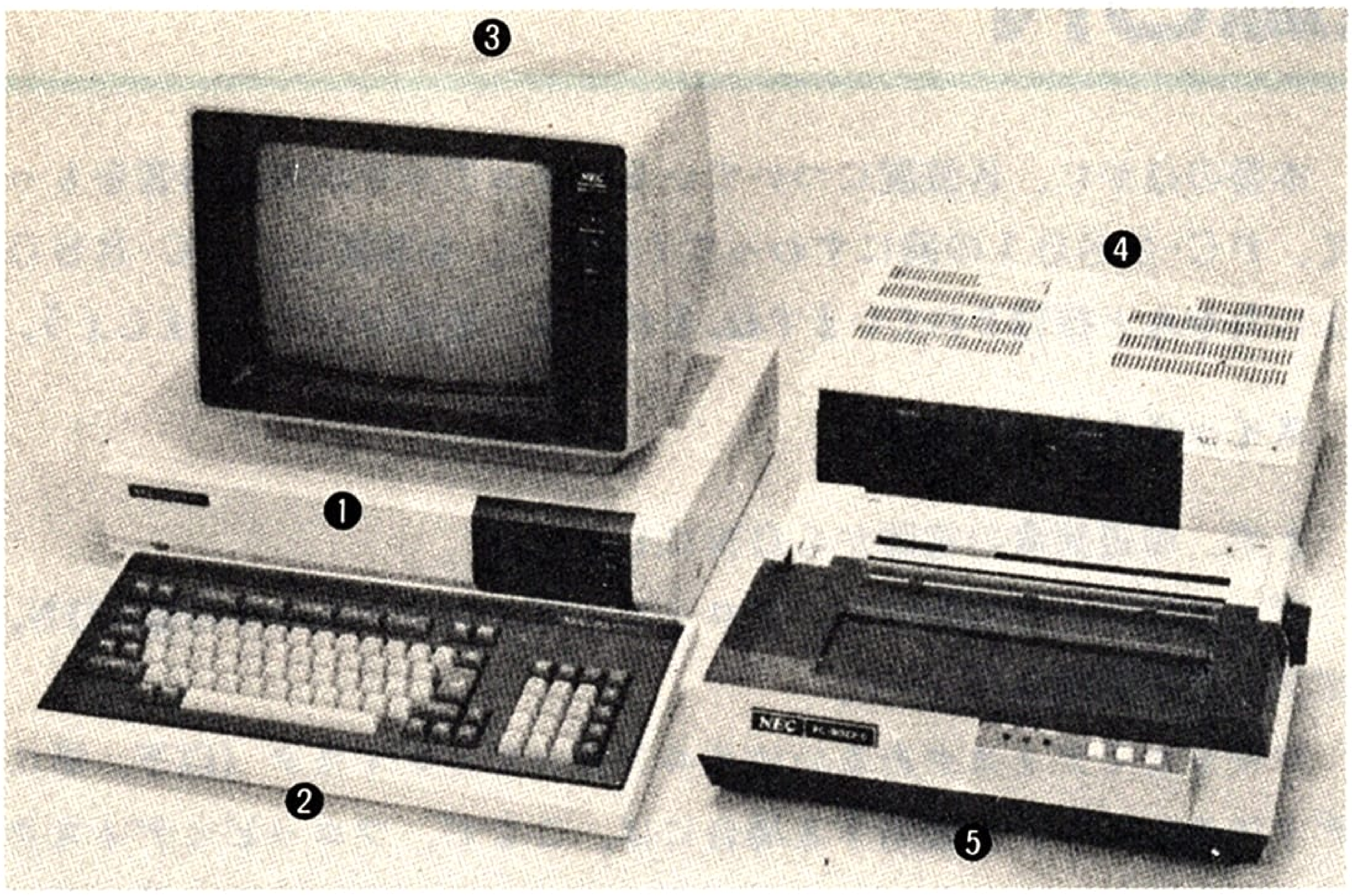
　ＰＣ−８８００シリーズでシステムを構成した例です。



❶　本　　体＊ ｝ＰＣ−８８０１
❷　キーボード ｝ＰＣ−８８０１
❸　１４インチ・カラー専用高解像度ディスプレイ（ＰＣ−８８５３）
❹　8インチ標準フロッピィディスクユニット（ＰＣ−８８８１）
❺　拡張用８インチ標準フロッピィディスクユニット（ＰＣ−８８８２）
❻　18ピン・ドットマトリックス漢字プリンタ（ＰＣ−８８２２）

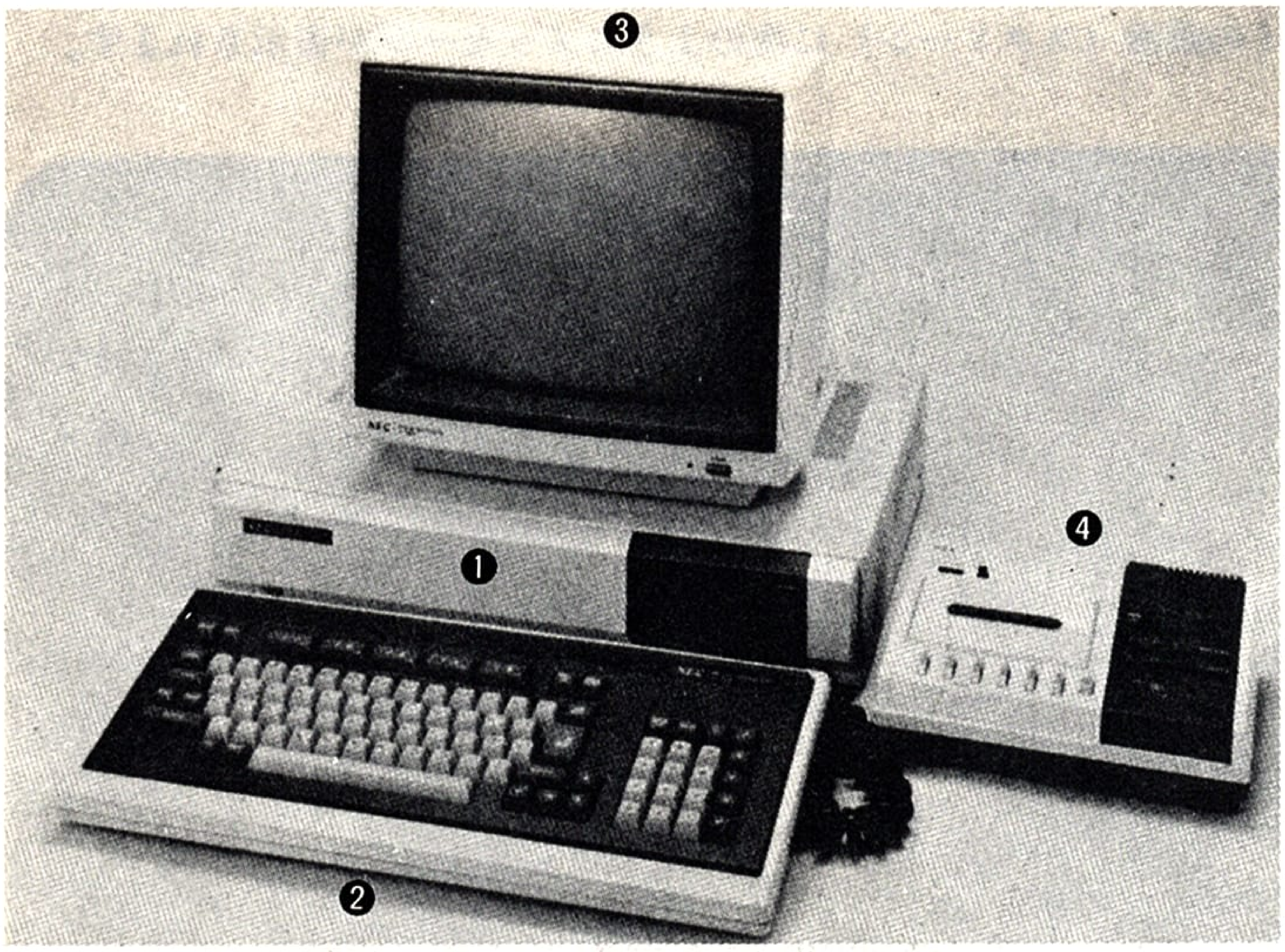
＊　本体には、ＰＣ−８８０１用漢字ＲＯＭボード（ＰＣ−８８０１−０１）が実装されています。

PC-8000シリーズの周辺装置を接続したシステム構成



❶ 本　　体 ⎫
❷ キーボード ⎭ PC-8801
❸ 12インチ・カラー高解像度ディスプレイ（PC-8049）
❹ 両面倍密度デュアルミニディスクユニット（PC-8031-2W）
❺ ドットマトリックスプリンタ（PC-8023-C）

最小システム構成



❶ 本　　体 ⎫
❷ キーボード ⎭ PC-8801
❸ 14インチ・モノクロ専用高解像度ディスプレイ（PC-8851）
❹ データ・レコーダ（PC-6082）

0・3　電源ON

本書のほかに、各装置についているマニュアルや取扱説明書をもって、ＰＣ-８８０１の前にすわってください。各装置は、ＰＣ-８８０１にちゃんと接続されていますね。それでは、電源をONにしましょう。

スイッチを入れる順序

周辺装置の電源ＯＮ

まず、周辺装置の電源からＯＮにします。

ディスプレイの電源ＯＮ、フロッピィディスクユニットの電源ON、プリンタの電源ＯＮ、……。

フロッピィディスクあるいはミニフロッピィディスク（以下、単にディスクという）を使用する場合には、ここで、フロッピィディスクユニットあるいはミニフロッピィディスクユニット（以下、単にディスクユニットという）のドライブ１にシステムディスクをセットします。

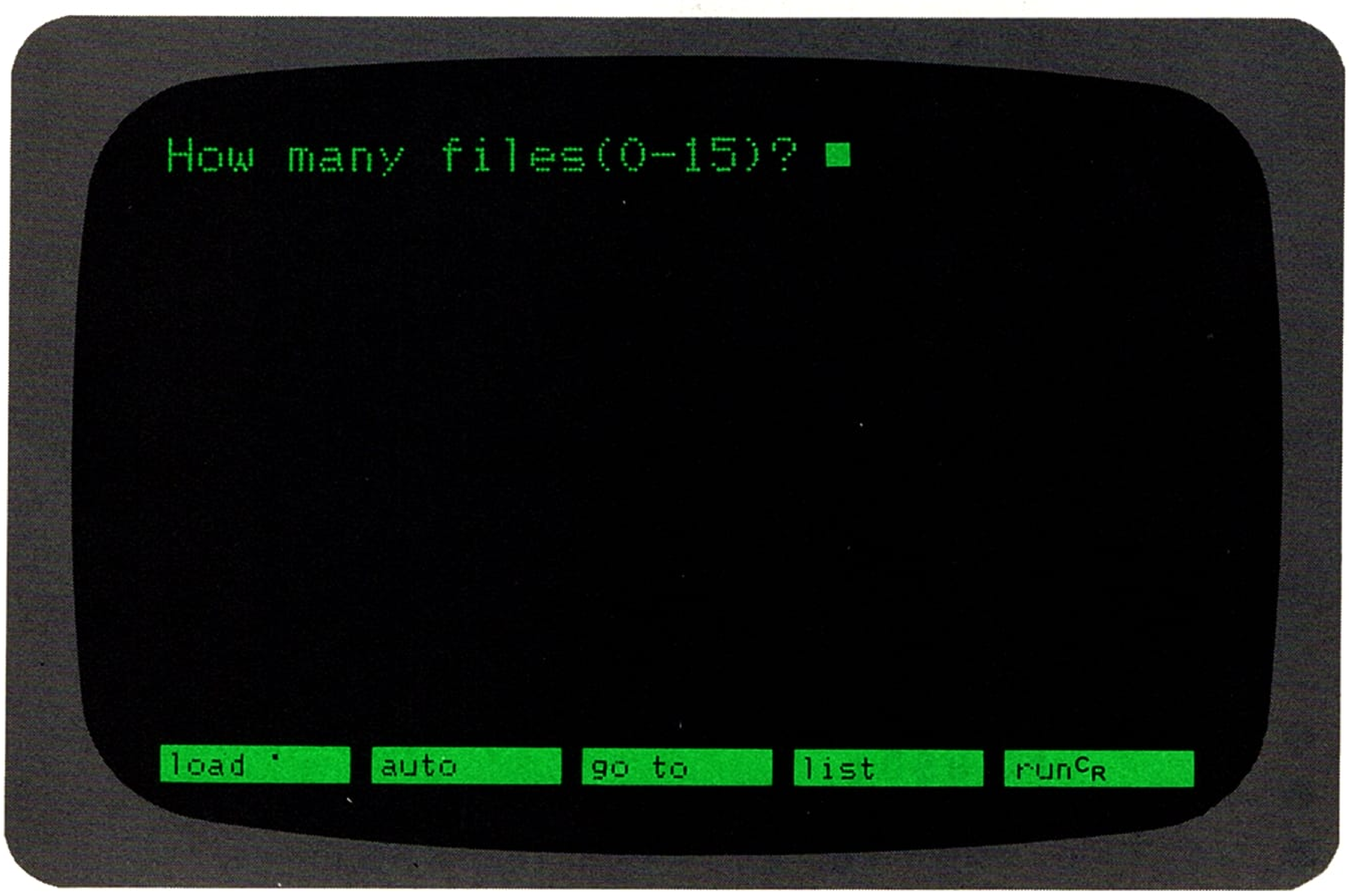
ＰＣ-８８０１の電源ＯＮ

最後に、ＰＣ-８８０１の電源をＯＮにします。いいですか。

ディスプレイの画面は？

メッセージ

ディスクを使用しない場合には、電源をＯＮにすると、ディスプレイの画面（スクリーン）に次のようにメッセージが表示されます。



なお、ディスプレイの明るさを調整する場合には、ディスプレイのBRIGHTツマミ（明るさ調整ツマミ）を回してください。

ここで、キーボードの⏎キー（リターンキー）を押します。なお、⏎キーは、キーボードに２つあります。まん中より少し右に大きなキーが１つ、それから右下に１つです。どちらを押してもかまいません。使いやすいほうを押しましょう。 ⏎キー（リターンキー）

すると、ディスプレイに次のようにメッセージが表示されます。



画面のOkの表示は、「準備Ok」というPC-8801からあなたへのメッセージです。 Ok

また、OkのOの文字の下で■がブリンク（点滅）していますね。この■をカーソルといいます。 カーソル

画面に上のようなメッセージが表示されない場合には、まず全装置の電源をOFFにします。そして、マニュアルや取扱説明書をみながら、全装置の接続の仕方をチェックしてください。また、PC-8801本体の背面にあるディップスイッチやジャンパスイッチを調べてください。

ディスクを使用する場合には、電源をONにして、PC-8801本体の前面左にあるリセットボタンを押します。 リセットボタン

すると、ディスプレイに次ページに示すようなメッセージが表示されます。



ここで、⏎キーを押します。画面は、次のようになります。



リセット

リセットボタン

ここで、ＰＣ-８８０１本体前面左にあるリセットボタンを押してみましょう。

ディスプレイの画面はどうなりましたか。

ディスクを使用しない場合は、次のようになります。



ディスクを使用する場合は、次のようになります。



どこかで見た画面ですね。そうです。電源をＯＮにしたときと同じ画面になりましたね。

このように、リセットボタンを押すと、ＰＣ-８８０１は電源をＯＮにしたときと同じ状態にリセットされるのです。

リセット

リセットボタン

しかし、むやみにリセットボタンを押してはいけません。

次の場合に限り、リセットボタンを使用します。

❶ 電源をＯＮにした直後

❷ キーボードから何もキーインできなくなった場合

❸ ディスクユニットにシステムディスクを読みに行かせる場合

電源をＯＦＦに

電源ＯＦＦ

いったん、電源をＯＦＦにしてみましょう。

ディスクを使用している場合には、まずディスクを取り出します。

ＰＣ-８８０１の電源をＯＦＦにします。

そして次に、周辺装置の電源をＯＦＦにします。ディスプレイの電源ＯＦＦ、ディスクユニットの電源ＯＦＦ、プリンタの電源ＯＦＦ、……。

もう一度、電源ＯＮ

電源ＯＮ

それでは、もう一度、電源をＯＮにしてみましょう。

電源をＯＮにする手順は、はじめは周辺装置、最後にＰＣ-８８０１の順序でしたね。

なお、ＰＣ-８８０１や周辺装置の電源をＯＦＦにしたときは、５秒以上たってから電源をＯＮにするようにしてください。

準備Ｏｋ、次はキー操作

Ｏｋ
カーソル

ディスプレイの画面にＯｋが表示されていれば、「準備Ｏｋ」でしたね。また、Ｏｋの下に■（カーソル）がブリンク（点滅）していますね。

それでは、次に、あなたの目の前にあるＰＣ-８８０１のキーボードのキーにさわってみることにします。

こんなに多くのキーをどう操作するのでしょうか。

心配しないで、とにかくさわってみてください。

キーを押しまちがえても、ＰＣ-８８０１がこわれることはありません。人差指１本でいいんです。どんどんキーを押してみてください。

キー操作に慣れることが必要です。

習うより慣れろです。

第1章

キー操作に慣れよう

◆

1・1　キーのいろいろ

どんなキーがありますか

あなたの目の前にあるＰＣ-８８０１のキーボード上のキーをざっと見てみましょう。

たくさんキーがありますね。

どんなキーがありますか。

アルファベットキー　**カタカナキー**　**数字キー**　**特殊記号キー**

まず、キーボードのまん中には、白色のキーがたくさん並んでいます。アルファベットキー（[A]、[B]、[C]、……[Z]）、カタカナキー（[ア]、[イ]、[ウ]、[エ]、[オ]、……、[ン]）、数字キー（[1]、[2]、[3]、……、[9]、[0]）、特殊記号キー（[+]、[−]、[*]、[／]、[?]、……）などのキーです。

特殊キー

次に、それらのまわりにこげ茶色のキーがとりまいています。特殊キー（[⏎]、(SHIFT)、[←]、[→]、[↑]、[↓]、……）です。

テンキー　**特殊キー**

そして、キーボードの右側に、白色およびこげ茶色のテンキーがあります。テンキーには、白色の数値入力用の [0]、[1]、[2]、[3]、……[9]キーのほかに、[+]、[−]のキーや特殊キー（(CLR)、(HELP)、[⏎]）があります。

グラフィックシンボルキー

以上のキーのほかに、キーボードのキーには直接示されていませんが、グラフィックシンボルキー（■、◪、[♥]、[♠]、……）もあります。

リピート機能

リピート機能

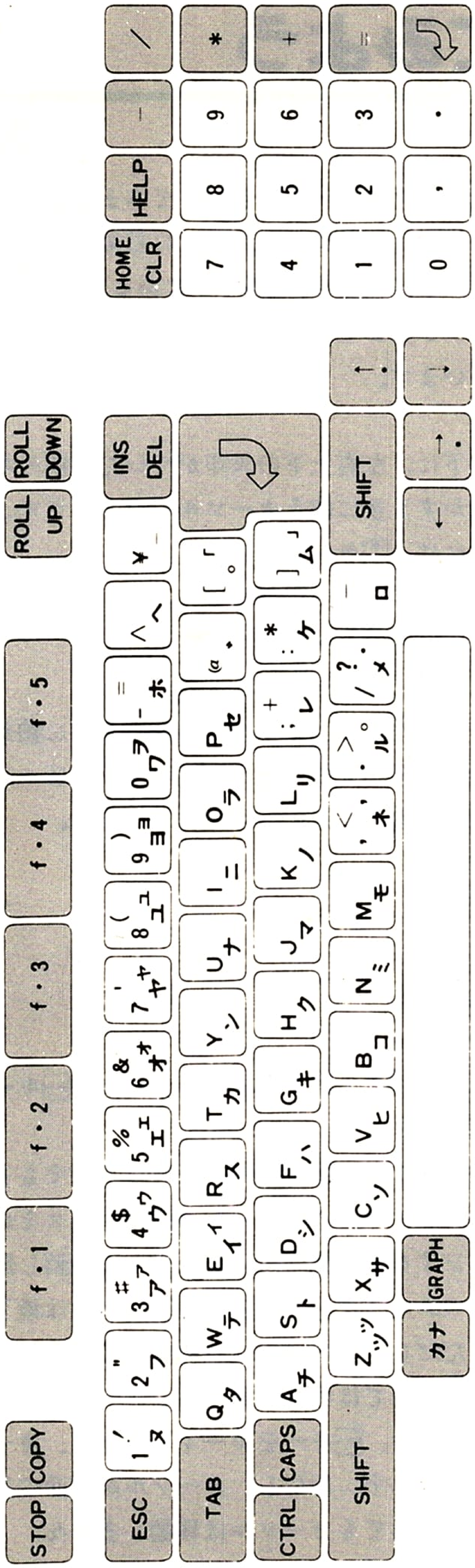
同じキーを約１秒間以上押し続けていると、同じ文字が連続してディスプレイの画面に表示されます。これをリピート機能といいます。

本書で使用しているキー表示

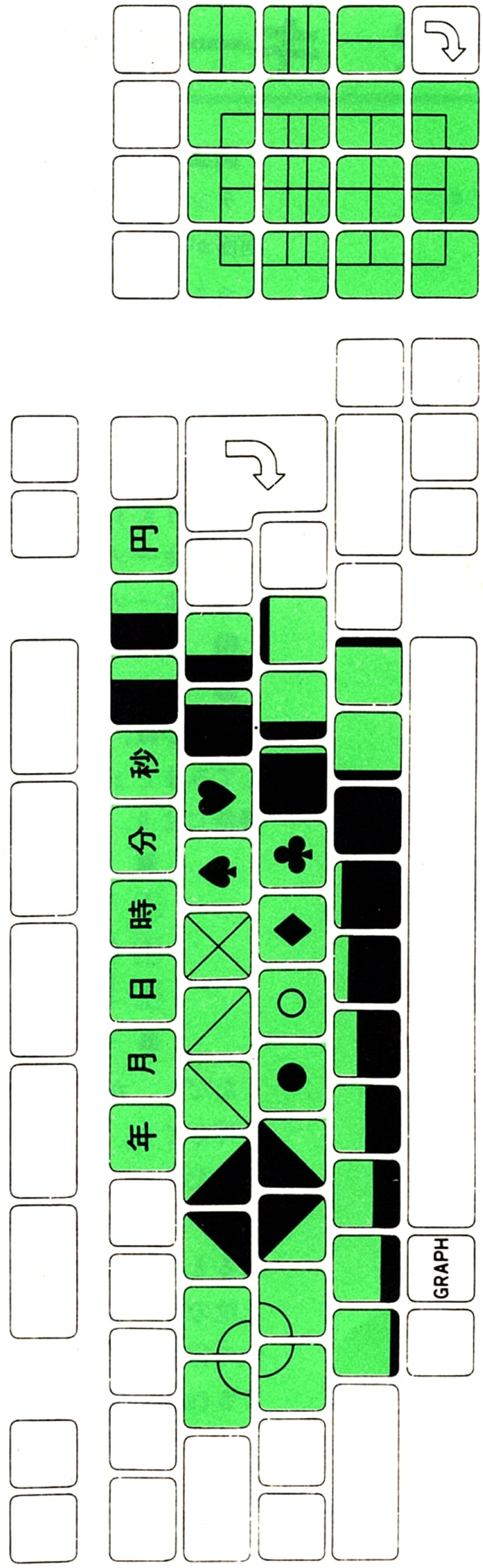
キーには、テンキーのほとんどのキーのように１文字しか表示していないキー（[0]、[1]、…）のほかに、２文字表示のキー（[A チ]、[Y ン]、…）、３文字表示のキー（[Z ッ ツ]、[2 " フ]、…）４文字表示のキー（[8 (ユ ュ]、[/ ? ・ メ]、…）があります。

本書でのキー表示

本書では、押すべき文字のみをキー表示することにします。たとえば、画面にＡと表示するのであれば、[A チ]キーではなく、[A]キーと表示しています。画面に"（ダブルクォテーションマーク）を表示するのであれば、[2 " フ]キーではなく、["]キーとキー表示しています。画面に？（クエッションマーク）を表示するのであれば、[/ ? ・ メ]キーではなく、[?]キーとキー表示しています。



キーの配列



グラフィックシンボルのキー配列

1・2　キーを押してみよう

画面をクリアしよう

CLR キー

テンキーの左上にある CLR（クリア）キーを押してみましょう。画面がクリアされ、カーソル（■）が画面の左上スミに移動します。もっとも、画面がクリアされるといっても、画面の下にある [load "]，……、[runCR] の表示はクリアされません。なお、画面の左上スミの位置をホームポジションといいます。

ホームポジション

カーソルを動かしてみよう

キーボードのまん中より右下に、左右上下の矢印がついた4個のキーがあります。カーソルを動かすときに使うカーソル移動キーです。

カーソル移動キー

[↑]キー

[←]キー

[→]キー

[↓]キー

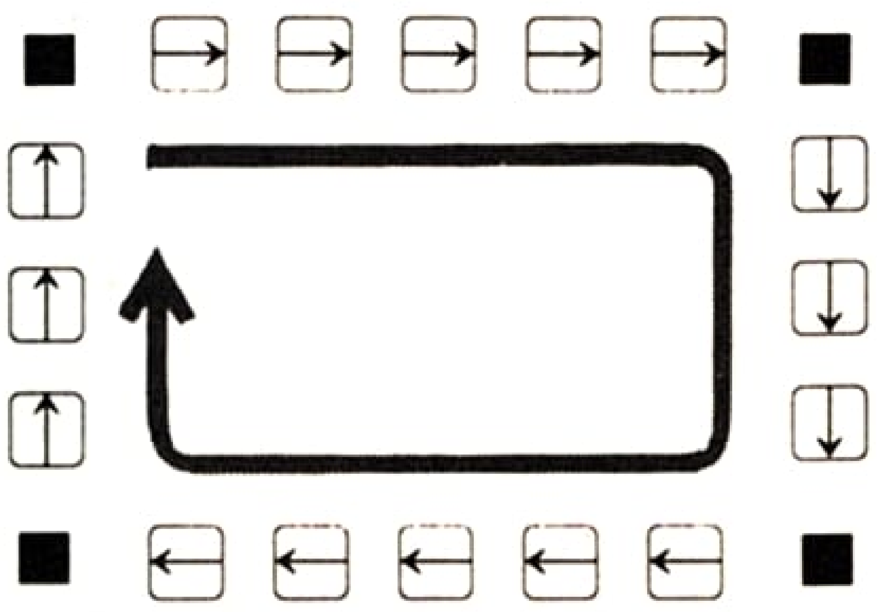
❶　カーソルを上に動かすには、[↑]キーを押します。

❷　カーソルを左に動かすには、[←]キーを押します。

❸　カーソルを右に動かすには、[→]キーを押します。

❹　カーソルを下に動かすには、[↓]キーを押します。

それでは、カーソル移動キーを押して、カーソルを左右上下に動かしてみましょう。なお、同じキーを約1秒以上押し続けて、リピート機能により、カーソルが連続して動くことを確かめてください。



ただし、カーソルが画面の左上スミにあるときは、[↑]キーや[←]キーを押しても、カーソルは移動しません。

またカーソルが画面の右端にあるときは、[→]キーをキーインすると、カーソルが次の行の左端に移動します。カーソルが画面の右下スミにあるときに、[→]キーをキーインすると、画面全体が1行上へ移動し最上位の行が消え（この動作をスクロールといいます）、カーソルは最下位の行の左端に移動します。ただし、この動作は、現在のようなクリアされた画面でのカーソル移動だけでは明確ではありません。

カーソルが画面の左端にあるとき、[←]キーをキーインすると、カーソルは1つ上の行の右端に移動します。ただし、カーソルが画面の左上スミにあるときは、[←]キーを押してもカーソルは移動しません。

カーソルが画面の最上位行にあるときには、[↑]キーをキーインしても、カーソルは移動しません。また、カーソルが画面の最下位行にあ

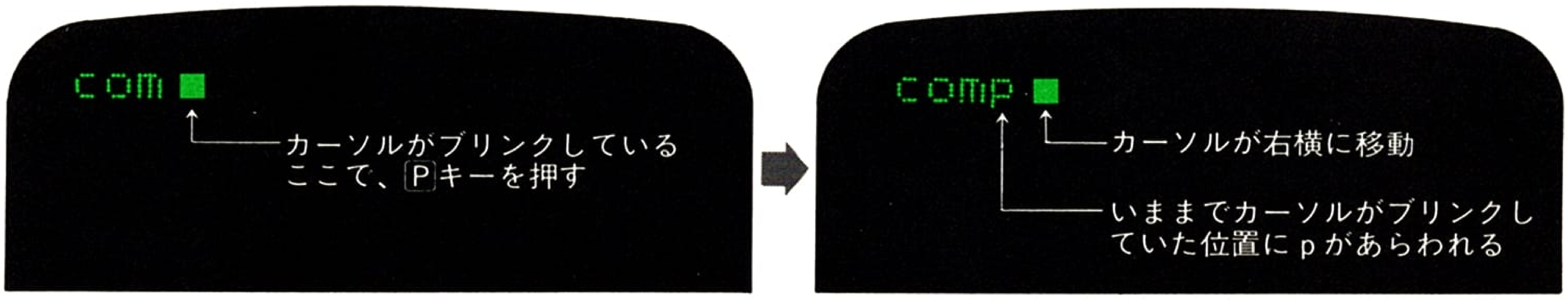
ときに、[↓]キーをキーインしても、カーソルは移動しません。

小文字のアルファベットを表示してみよう

アルファベット（小文字）

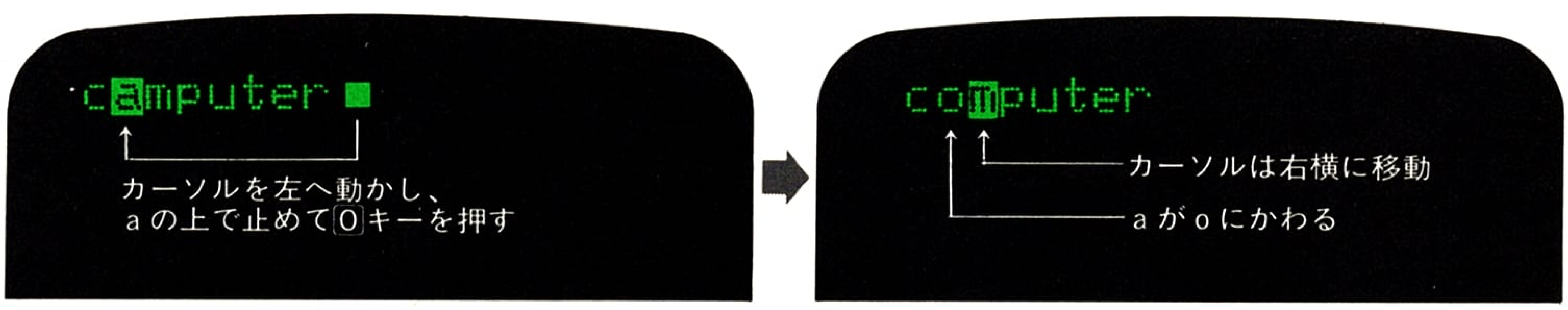
[A]、[B]、[C]、……、[Z]などのアルファベットキーを押してみましょう。小文字のアルファベットが画面に表示されます。ただし、アルファベットキーは、アルファベット順にはならんでいません。

画面をよく見ると、いままでカーソルがブリンク（点滅）していた位置に小文字のアルファベットがあらわれ、カーソルは右横に移動することがわかります。



カーソル

カーソルは、文字を表示する位置を示しています。カーソルを動かして、文字を表示する位置を自由に決めることができます。まちがった文字を修正したい場合には、カーソル移動キーを押してカーソルを動かし、修正したい文字の上で止め、正しい文字のキーを押せば正しい文字に置きかわります。



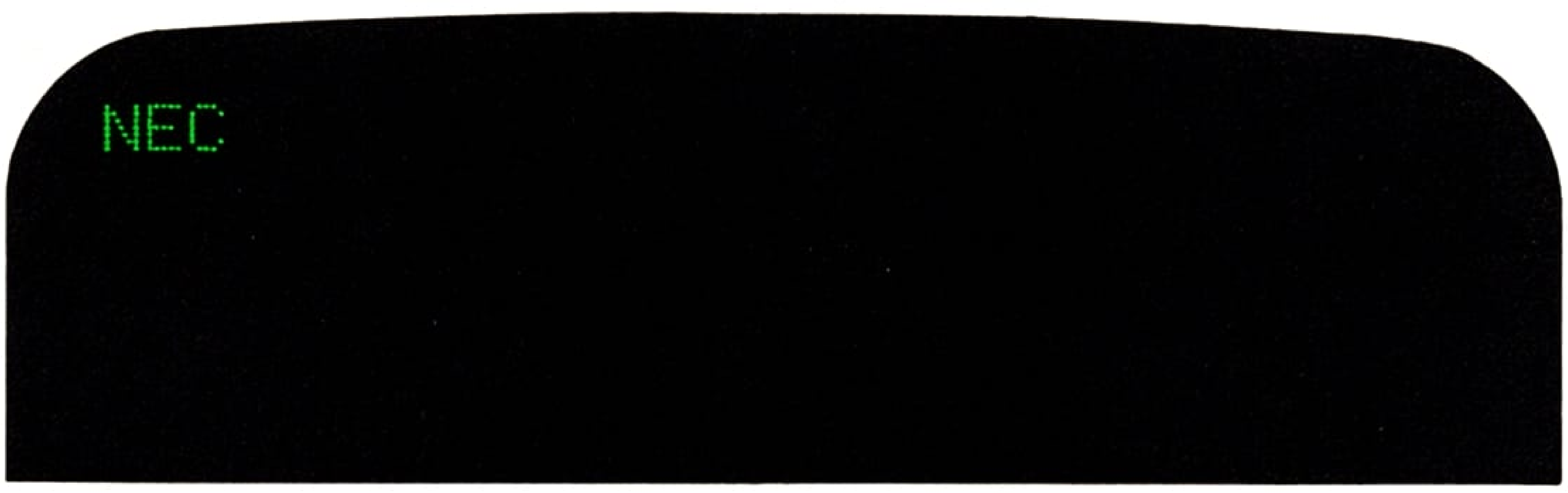
それでは、カーソル移動キーを押してカーソルを動かし、画面のあちこちに小文字のアルファベットを表示してみましょう。

大文字のアルファベットを表示するには

(SHIFT)キー

アルファベット（大文字）

(SHIFT)（シフト）キーを押しながらアルファベットキーを押すと、大文字のアルファベットが表示されます。やってみましょう。



SHIFTキー

なお、SHIFTキーは、キーボードの左下とまん中右下に１つずつあります。どちらを使っても同じですから、使いやすい方のキーを使いましょう。

CAPSキー

キーボード左下のCAPS(キャピタルロック)キーを押すと、キーはロックされます。この状態で、アルファベットキーを押せば、大文字のアルファベットが表示されます。このとき、SHIFTキーを押しながらアルファベットキーを押せば、小文字のアルファベットが表示されます。ロックされたCAPSキーをもう一度押すと、CAPSキーのロックは解除されます。ためしてみましょう。

カーソルを画面の左上スミに動かそう

HOMEキー

ここで、SHIFTキーを押しながら、テンキーの左上にあるHOME(ホーム)キーを押してみよう。カーソルが画面の左上スミに移動します。なお、画面の左上スミの位置をホームポジションといいます。

HOMEキーは、CLRキーと同じキーです。使い方のちがいがわかりますか。単にカーソルを画面の左上スミに動かすのであれば、SHIFTキーを押しながらHOMEキーを押します。画面をクリアしてカーソルを画面の左上スミに動かすには、CLRキーを押します。よろしいですね。

カナを表示するには

カナキー
カナモード
カナ文字
ノーマルモード

カナ(カナ)キーを押すと、キーはロックされ、カナモードになります。カナモードでは、各キーの下あるいは右に示されているカナ文字を画面に表示することができます。カナキーをもう一度押せば、ノーマルモードにもどります。ノーマルモードとは、キーボードでの普通のキーインをする状態をいい、アルファベット、数字、特殊記号などを画面に表示することができます。

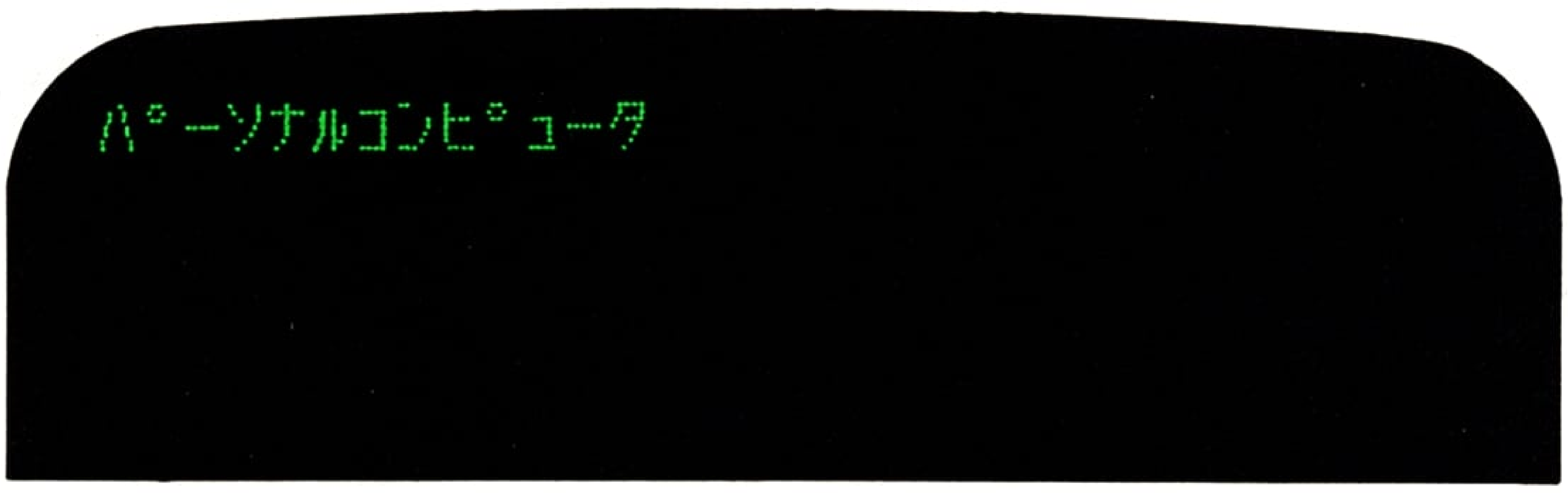
ここで、カナキーを押しカナモードにして、ア、イ、ウ、エ、オ、……、ンのキーを押し、カナ文字を画面に表示してみましょう。

キーボードのまん中右の大きな⏎(リターン)キーの左横の２つのキー゛と゜は、濁点（゛）と半濁点（゜）を示しています。ガギグゲゴ、パピプペポなどを画面に表示してみましょう。

キーボードなどの音びき（ー）を画面に表示するためには、大きな⏎キーの左上にあるーキーを使います。

カナ文字には、９つのカナ小文字（ァ、ィ、ゥ、ェ、ォ、ッ、ャ、ュ、ョ）がありますが、これを画面に表示するには、SHIFTキーを押しながら、ア、イ、ウ……、ヨのキーを押します。

カナ文字はアイウエオ順にならんでいないので、なかなか見つからないカナもあるでしょうが、がんばって探してください。



スペースをあけるには

スペース

アルファベットやカナを画面に表示するときに、スペースをあける必要が出てきます。スペースをあけるには、キーボードの一番下の長い［　　　　］（スペース）キーを押します。適当にスペースをあけながら、アルファベットやカナを画面に表示してみましょう。

［　　　　］キー

なお、本書では、スペースを１つあけることを△で表わすことにします。

△



数字を表示しよう

数字を画面に表示するには、キーボードの上部にある[1]、[2]、[3]、……、[9]、[0]のキーを押せばよいのですが、キーボード右側の数値入力用のテンキーを使うと便利です。テンキーを押してみましょう。

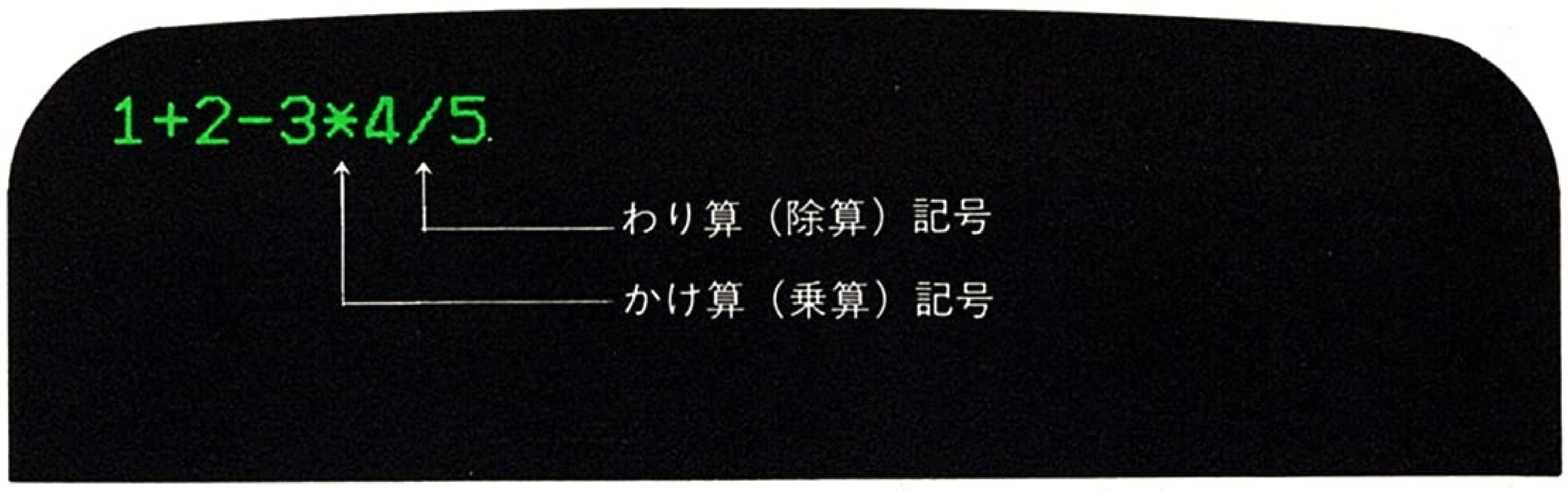
テンキー

演算記号を表示するには

次の演算記号のキーを押してみましょう。

[+]、[−]、[*]、[/]、[=]

[+]（プラス）はたし算（加算）記号、[−]（マイナス）はひき算（減算）記号、[*]（アスタリスク）はかけ算（乗算）記号、[/]（スラッシュ）はわり算（除算）記号、[=]（イコール）は等号です。これらのキーは、すべてテンキーのところにありますね。



その他のキーも押してみよう

キーボードの白色のキーのうち、まだ押していないキーがあります。どんどん押してみましょう。たとえば、キーボードまん中右下の[?]（クエッションマーク）キーを[SHIFT]キーを押しながら押してみましょう。また、その上にある[;]（セミコロン）キー、[:]（コロン）キーを押してみましょう。そして、キーボード左上の["]（ダブルクォテーションマーク）キーを[SHIFT]キーを押しながら押してみましょう。テンキーのところにある[,]（コンマ）キーも押してみましょう。

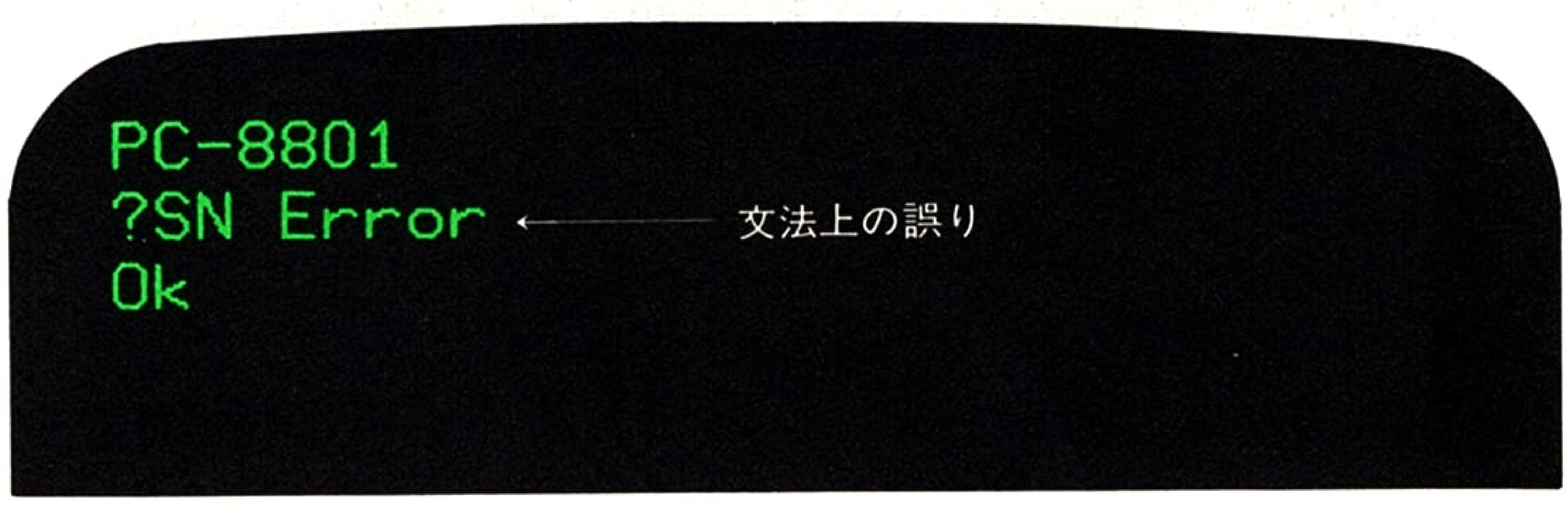
リターンキーを押すと

⏎キー（リターンキー）

エラーメッセージ

それでは、画面をクリアしてください。[CLR]キーを押せばよかったですね。次に、ＰＣ-８８０１とキーインして⏎（リターン）キーを押してみましょう。ピーとブザーの音がして、画面にエラーメッセージが表示されます。

ディスクを使用しない場合は、次のようになります。



ディスクを使用する場合は、次のようになります。



?SN Error

Syntax error

画面に「？ＳＮ　Ｅｒｒｏｒ」あるいは「Ｓｙｎｔａｘ　ｅｒｒｏｒ」が表示されましたね。これは、「文法上の誤りがあります」という

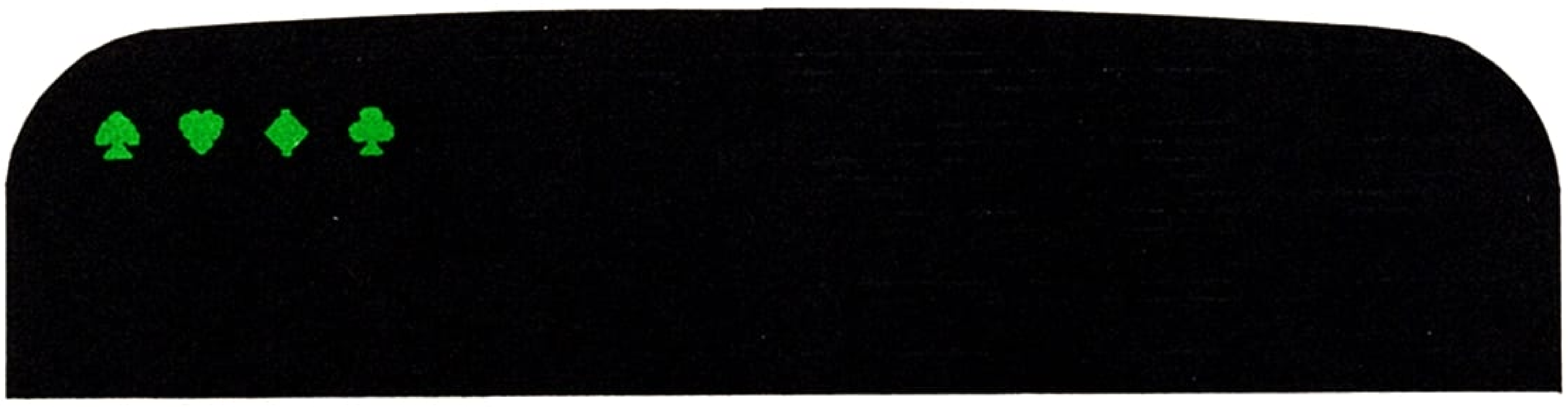
PC-8801からあなたへのエラーメッセージです。

「文法上の誤り」とは、何でしょうか。「?SN Error」あるいは「Syntax error」が出たら、どうしたらよいのでしょうか。心配ですね。しかしここでは、あまり気にしないでとにかく先へ進むことにしましょう。

グラフィックシンボルを表示するには

GRAPH キー

GRAPH(グラフィック)キーを押しながら、I、O、J、Kのキーを押してみましょう。♠、♥、♦、♣のマークが表示されましたね。



グラフィックシンボル入力モード

このように、GRAPHキーを押しながら、グラフィックシンボルをキー入力する状態をグラフィックシンボル入力モードといいます。

このグラフィックシンボルは、キーボードのどのキーを見ても書かれていません。別売のグラフィックラベルを入手して、それぞれがいとうするキーにはっておくとよいでしょう。

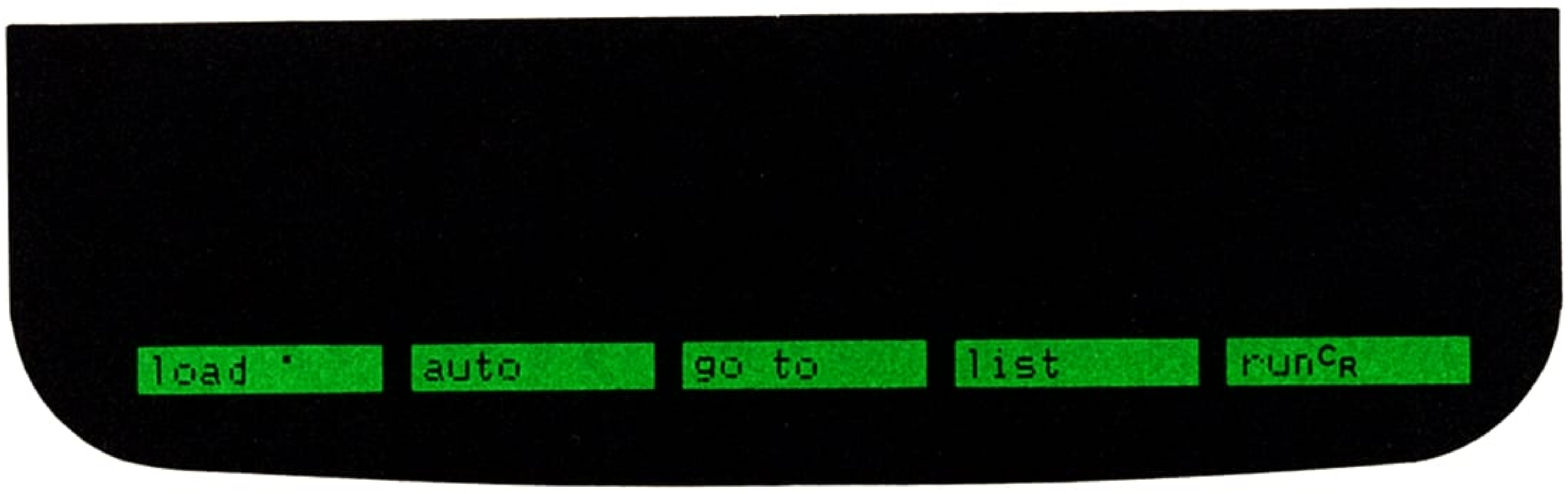
なお、グラフィックシンボルキーでないキーを押しても、何も表示されません。念のため。

それでは、いろいろなグラフィックシンボルを表示する練習をしてみてください。

ファンクションキーとは何でしょうか

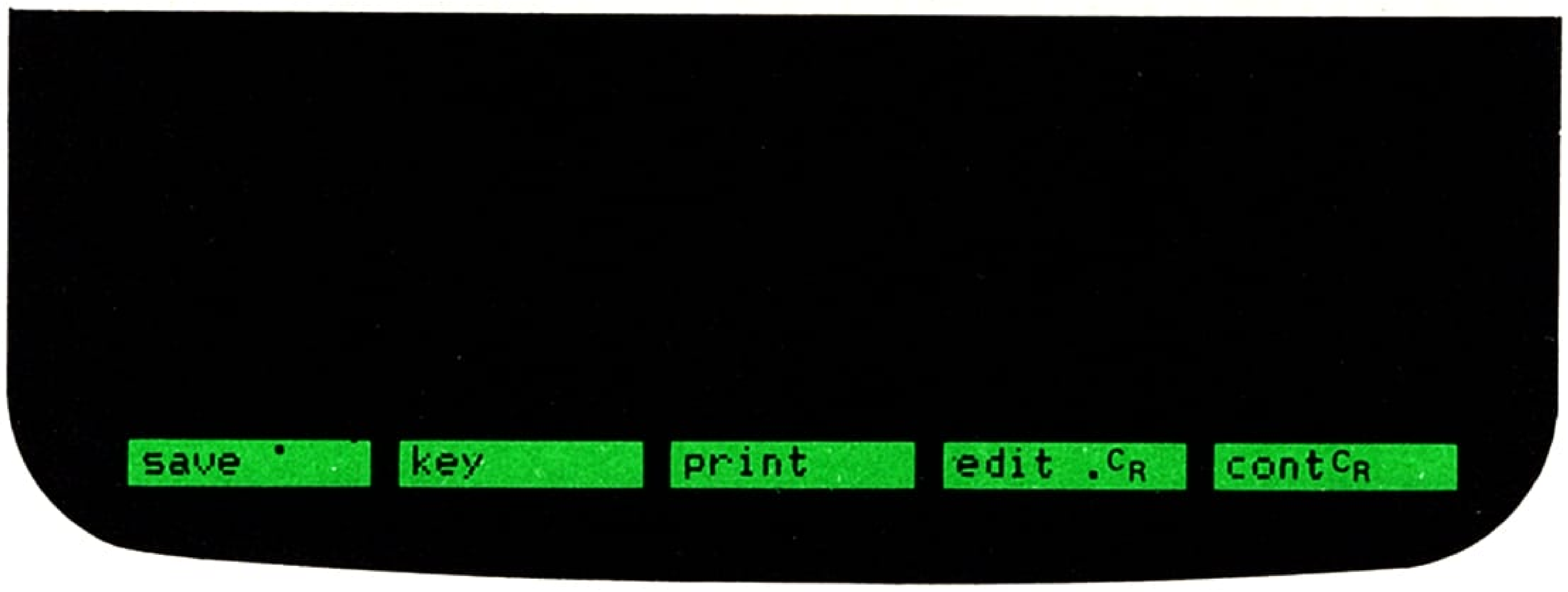
ファンクションキー

キーボードの一番上に、f·1からf·5までのこげ茶色のキーが5つ並んでいます。これらのキーをファンクションキーと呼びます。画面を見てください。



画面の一番下に白黒が反転している行がありますね。その中に表示されている文字がそれぞれ、ファンクションキー f·1 から f·5 までに定義されている内容です。

次に、 (SHIFT) キーを押した状態の画面を見てください。



表示されている文字が変わりましたね。

(SHIFT) キーを押した状態では、 (f·1) キーのｆ·1をｆ·6と考え、 (f·2) キーのｆ·2をｆ·7と考え、 (f·5) キーのｆ·5をｆ·10と考えます。したがって、画面に表示されている文字が、それぞれ、 (SHIFT) キーを押した状態のファンクションキー (f·1) から (f·5) までに定義されている内容です。

ＰＣ-８８０１の電源をＯＮにしたとき、あるいはリセットボタンを押したときには、常に、ファンクションキーは次のように定義されています。

f・1	load△" ← △はスペースを１つあけることを意味します
f・2	auto△
f・3	go△to△
f・4	list△
f・5	run$^{C}_{R}$ ←
f・6	save△"
f・7	key△
f・8	print△
f・9	edit.△$^{C}_{R}$ ←
f・10	cont$^{C}_{R}$ ←

$^{C}_{R}$は⏎キーを押すのと同じ

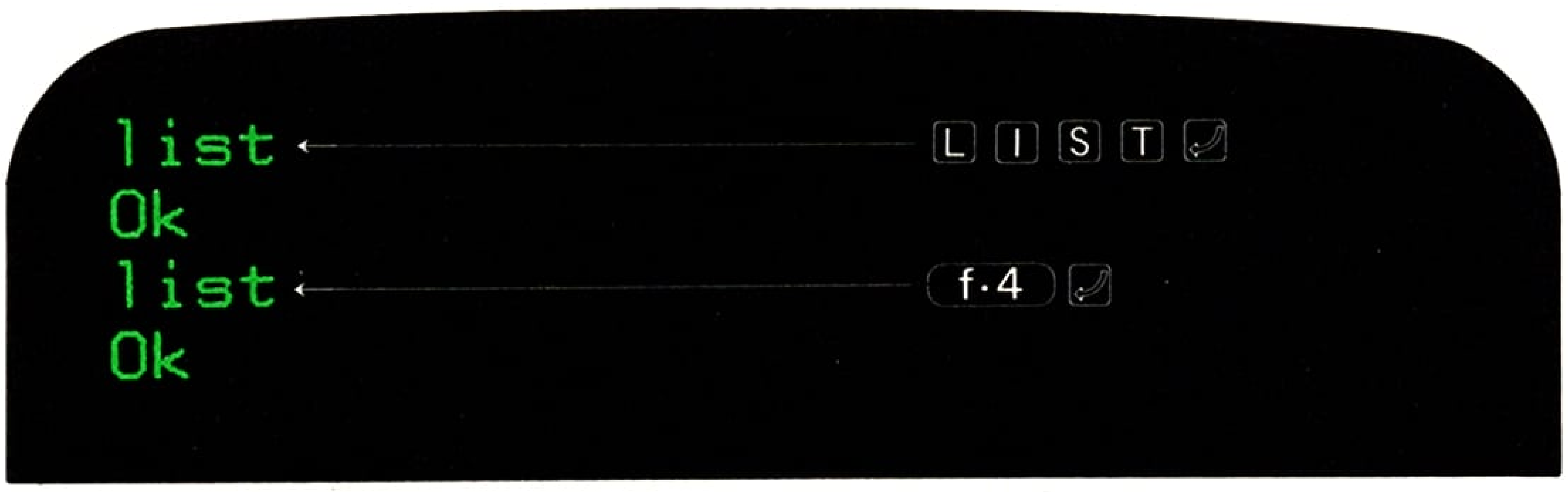
（ｆ·6～ｆ·10は (SHIFT) キーを押した場合です）

なお、 $^{C}_{R}$ は⏎キーを押すのと同じ機能を表します。

定義内容の具体的意味はさっぱりわかりませんね。それでかまいません。とにかく、先へ進みましょう。

次のようにキーインしてみてください。画面はどうなりましたか？

list ⏎	← L I S Tとキーを押し、⏎キーを押す
(f·4)	← つづいて (f·4) キーを押し、⏎キーを押す



どちらも同じ結果になりましたね。

このように、ファンクションキーを押すと、定義内容がキーから入力されたのと同じになります。たとえば、(f·4)キーを１回押すだけで、いっぺんに４文字分（ｌｉｓｔ）をキーボードから押したのと同じになります。便利ですね。

ファンクションキーの定義内容を画面に表示するには、次のようにキーインします。ＫＥＹ△ＬＩＳＴ(キーリスト)です。なお、ＫＥＹ△ＬＩＳＴは、小文字のｋｅｙ△ｌｉｓｔでもかまいません。

KEY LIST

KEY△LIST ⏎ ←ファンクションキーの定義内容を画面に表示
↑ スペースを１つあける



ところで、上の画面からもわかるように、ｋｅｙは(f·7)キーに定義されており、ｌｉｓｔは(f·4)キーに定義されていましたね。そこで、ファンクションキーの定義内容を調べるには、次のようにキーを押せばよいのです。やってみましょう。

(f·7) (f·4) ⏎ ←ファンクションキーの定義内容を画面に表示

なお、(f･7)キーを押すと、ｋｅｙと画面に表示したあとに、自動的にスペースを１つあけます。同じように、(f･4)キーを押すと、ｌｉｓｔと画面に表示したあとに、自動的にスペースを１つあけます。

ＫＥＹ　ＬＩＳＴは、「ファンクションキーの定義内容を画面に表示せよ」、とＰＣ-８８０１に命令しているのです。このように決められた内容をＰＣ-８８０１に命令するという意味をもつ言葉をコマンドといいます。なお、コマンドは大文字でも小文字でもかまいません。

コマンド

したがって、ファンクションキーの定義内容を画面に表示するためには、ＫＥＹ　ＬＩＳＴコマンドを使えばいいのです。

KEY LIST コマンド

さて、ファンクションキーは自由に定義して使うことができます。ファンクションキーの内容を定義するには、ＫＥＹ（キー）文を使って、次のようにキーイン（小文字でもよい）します。

KEY文

```
KEY △キー番号 ," 定義内容 " ⏎   ← ファンクションキーの定義
    ↑
    └─ スペースを１つあける
```

たとえば、(f･3)キーにｉｎｐｕｔ△を定義するには、次のようにします。

```
key△3,"input△" ⏎   ← (f･3)キーにｉｎｐｕｔ△を定義
```

しかし、(f･7)キーの定義内容がｋｅｙ△でしたから、次のようにキーインしましょう。

```
(f･7) 3,"input△" ⏎   ← (f･3)キーにｉｎｐｕｔ△を定義
```

ファンクションキーの内容を定義する場合、⏎キーを押すのと同じ働きをする C_R を定義するには、次のようにキーインします。

```
key △キー番号 ," 定義内容 "+chr$(13) ⏎
                           └──┬──┘
                              └─ CR の定義
```

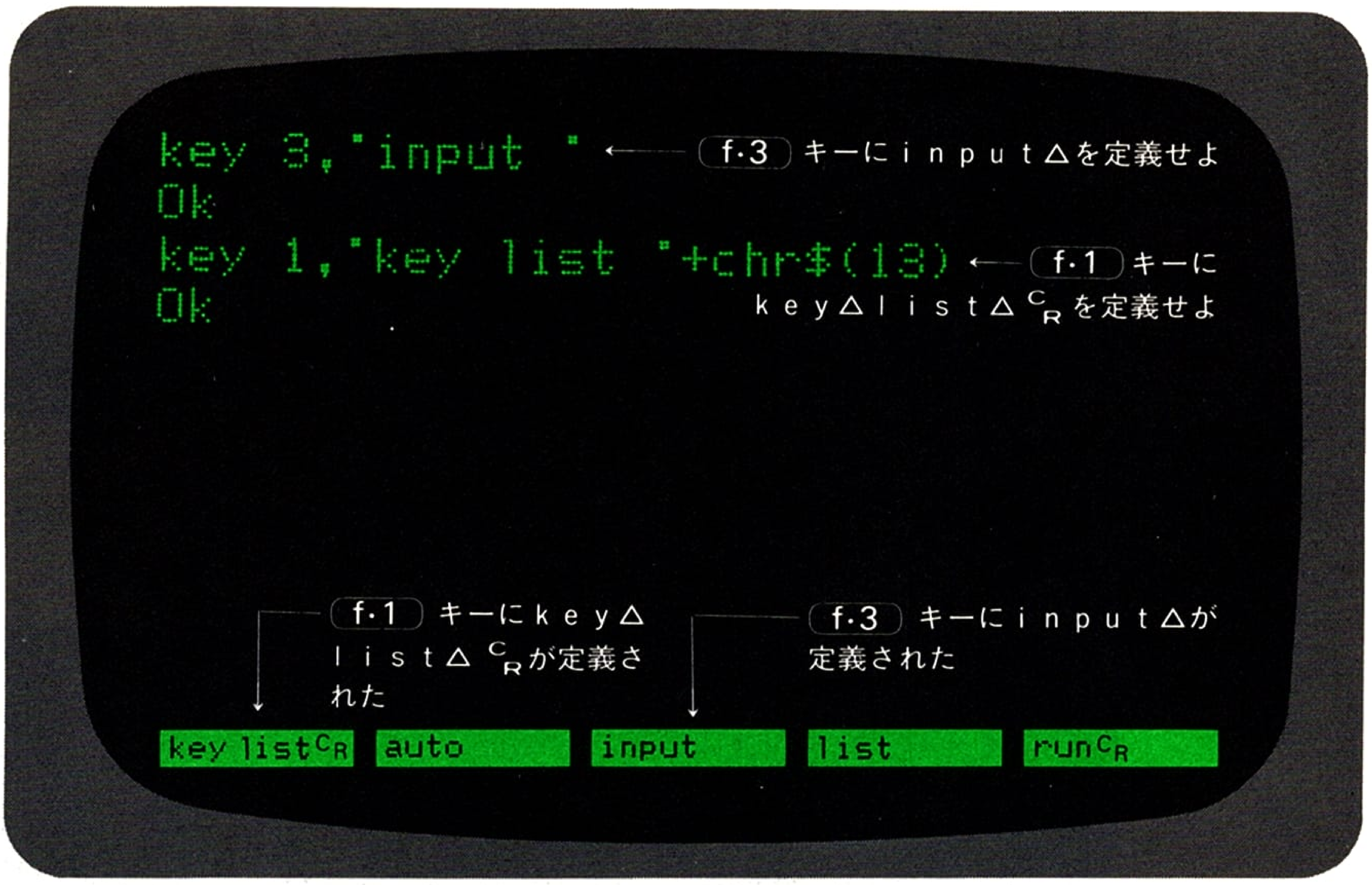
たとえば、(f･1)キーにｋｅｙ△ｌｉｓｔ△ C_R を定義するには、次のようにします。

```
key△1,"key△list△"+chr$(13) ⏎   ← (f･1)キーに
                                    ｋｅｙ△ｌｉｓｔ△CR を定義
```

次のようにキーインしてみましょう。

```
(f･7) 1," (f･7)   (f･4)  "+chr$(13) ⏎
  ||       ||      ||
 key△     key△   list△
```

画面は次ページのようになります。



画面の一番下の行を見てください。f·1 キーにkey△list△^C_R が定義され、f·3 キーにinput△が定義されたことがわかりますね。なお、f·1 キーに定義されていたload"△や f·3 キーに定義されていたgo△to△は消えてしまいます。

それでは、ファンクションキーの定義内容を画面に表示してみましょう。いま、f·1 キーにkey△list△^C_R を定義しましたから、f·1 キーを押すだけでよいのです。

f·1 ←——— ファンクションキーの定義内容を画面に表示

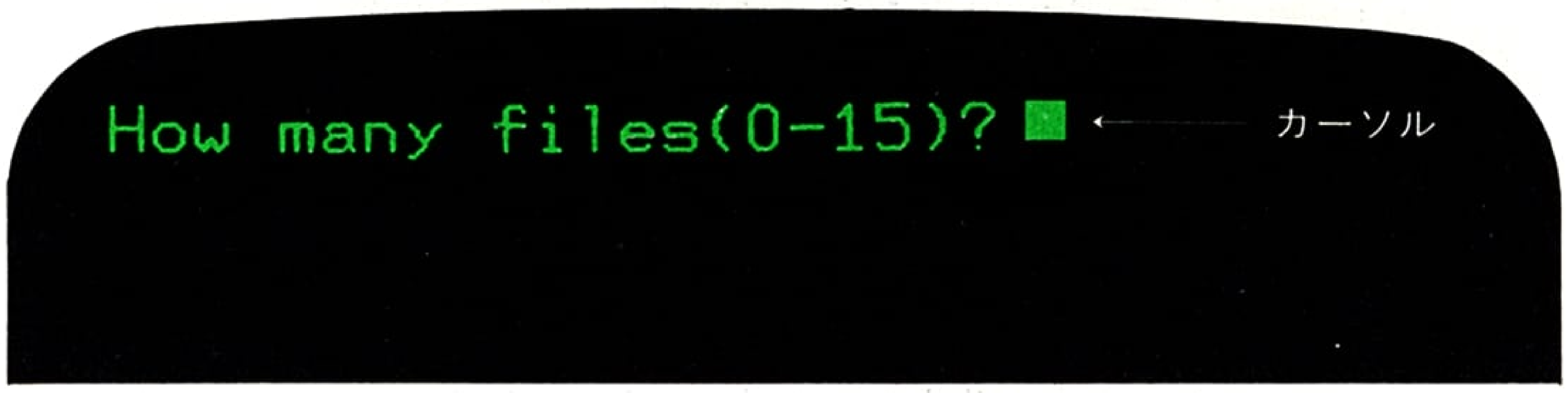
画面は次のようになります。



この画面からも、f·1 キーにkey△list△^C_R が定義され、f·3 キーにinput△が定義されたことがわかりますね。

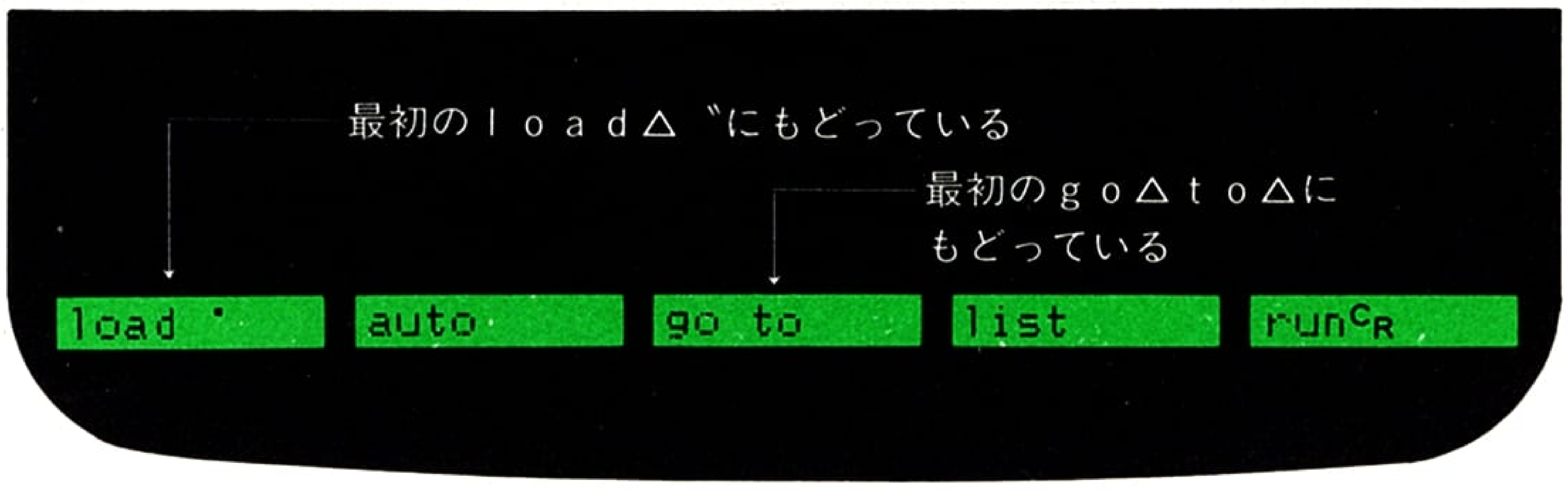
ここで、いったん電源をOFFにするかリセットボタンを押してみ

ましょう。電源をＯＦＦにした場合は、しばらくしたら、もう一度電源をＯＮにします。画面に次のメッセージが出ます。



次に、⏎キーを押します。すると、Ｏｋが表示されますね。

画面の一番下のファンクションキーの定義内容の表示は、どうなりましたか。

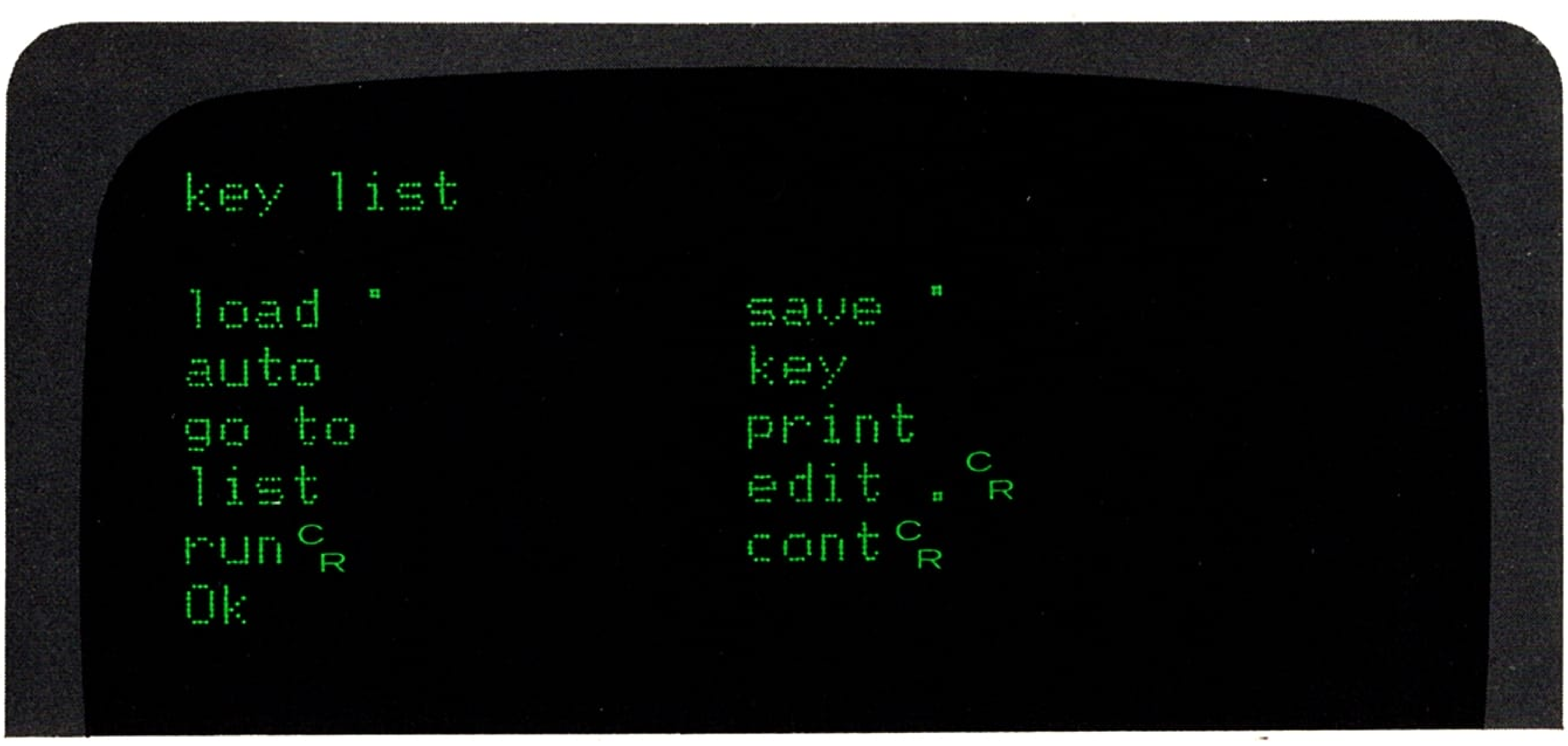


f·1 キーは、ｋｅｙ△ｌｉｓｔ△ $^{C}_{R}$ から最初のｌｏａｄ△"にもどっています。また、f·3 キーも、ｉｎｐｕｔ△から最初のｇｏ△ｔｏ△にもどっています。

このことを、ファンクションキーの定義内容を画面に表示して確かめてみましょう。次のようにキーインしてください。

f·7　f·4　⏎ ← ファンクションキーの定義内容を画面に表示

画面は次のようになります。



このように、ファンクションキーは自由に定義して使うことができますが、電源をＯＦＦにするかリセットボタンを押せば、その定義内容は消えてしまいます。この点、注意が必要です。

1・3　キーを押しまちがえたら

どのキーを使うか

キーを押しまちがえた場合には、次のキーを使って修正します。

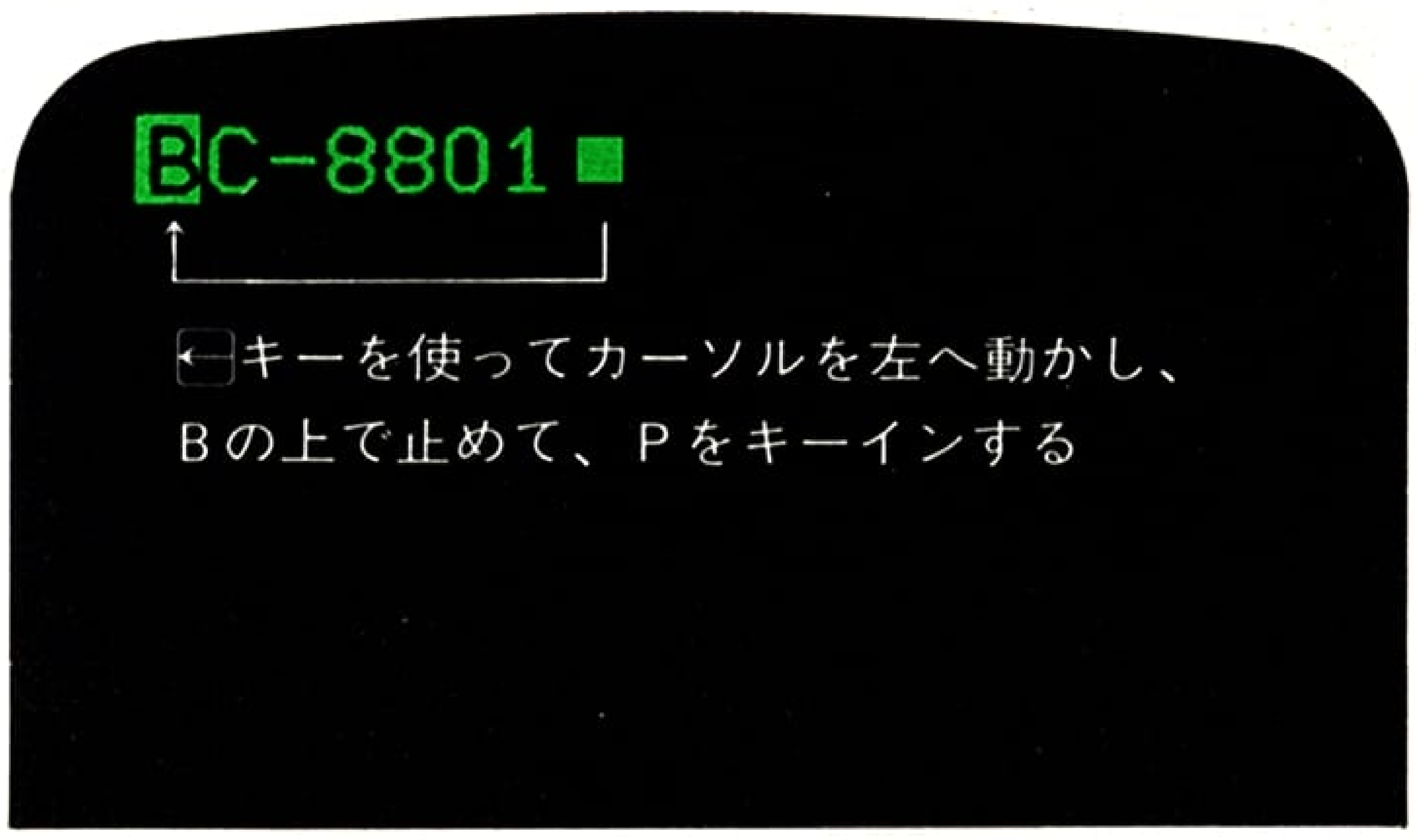
❶ ⇧……カーソルを１行上へ動かす。　⇧キー
❷ ⇦……カーソルを１桁左へ動かす。　⇦キー
❸ ⇨……カーソルを１桁右へ動かす。　⇨キー
❹ ⇩……カーソルを１行下へ動かす。　⇩キー
❺ (DEL)…カーソルの左の文字を１文字デリート（削除）し、カーソルより右の文字を左へつめます。　(DEL)キー
❻ (INS)…カーソルがブリンク（点滅）している文字とその前の文字との間に文字をインサート（挿入）できます。なお、(INS)キーだけは、(SHIFT)キーを押しながら押します。　(INS)キー

それでは、カーソル移動キー（⇧、⇦、⇨、⇩）(DEL)（デリート）キー、(INS)（インサート）キーを使って、キーを押しまちがえた場合の修正をしてみましょう。

まちがった文字を修正するには

まちがった文字を修正したい場合には、⇧、⇦、⇨、⇩キーを使ってカーソルを動かし、修正したい文字の上で止めて、正しい文字を入力すれば新しい文字にかわります。

たとえば、ＰＣ-８８０１をまちがってＢＣ-８８０１とキーインしたとしましょう。ＢをＰに修正するには、⇦キーを使ってカーソルを左へ動かし、Ｂの上で止めて、Ｐを入力します。



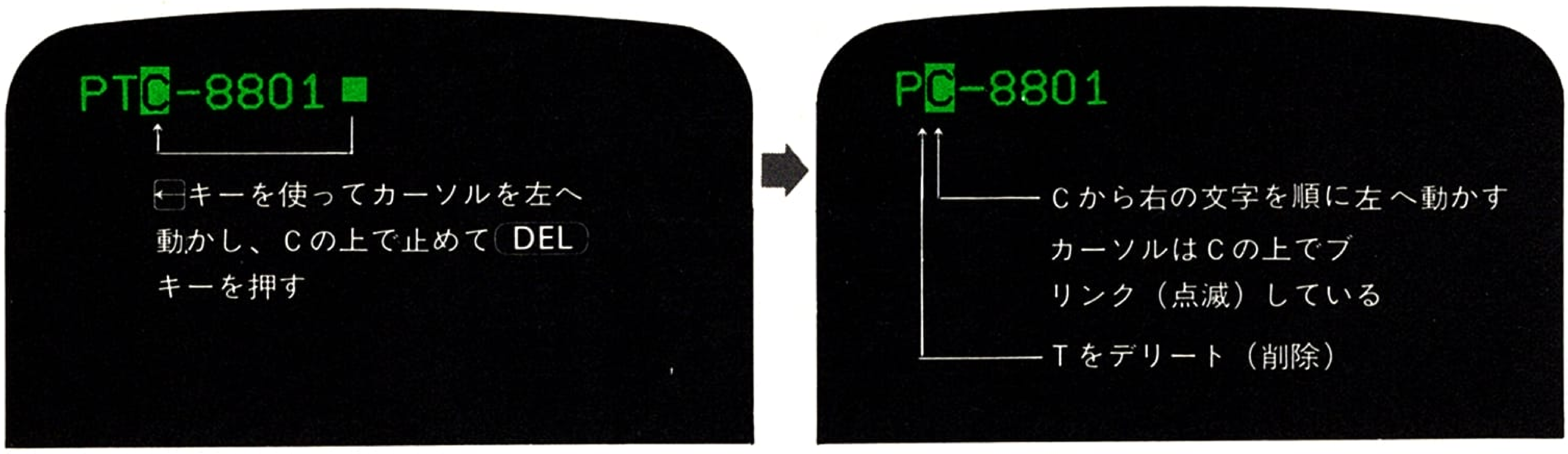


文字をデリート（削除）するには

文字をデリート（削除）したい場合には、その文字のすぐ右側まで

カーソルを動かし、DELキーを押します。デリートしたい文字の個数だけDELキーを押すと、押した回数だけカーソルの左側の文字をデリートし、カーソルの右側の文字を順に左へ動かします。

たとえば、ＰＣ-８８０１をまちがって、ＰＴＣ-８８０１とキーインしたとしましょう。Ｔをデリートするには、←キーを使ってカーソルを左へ動かし、Ｃの上で止めて、DELキーを押します。

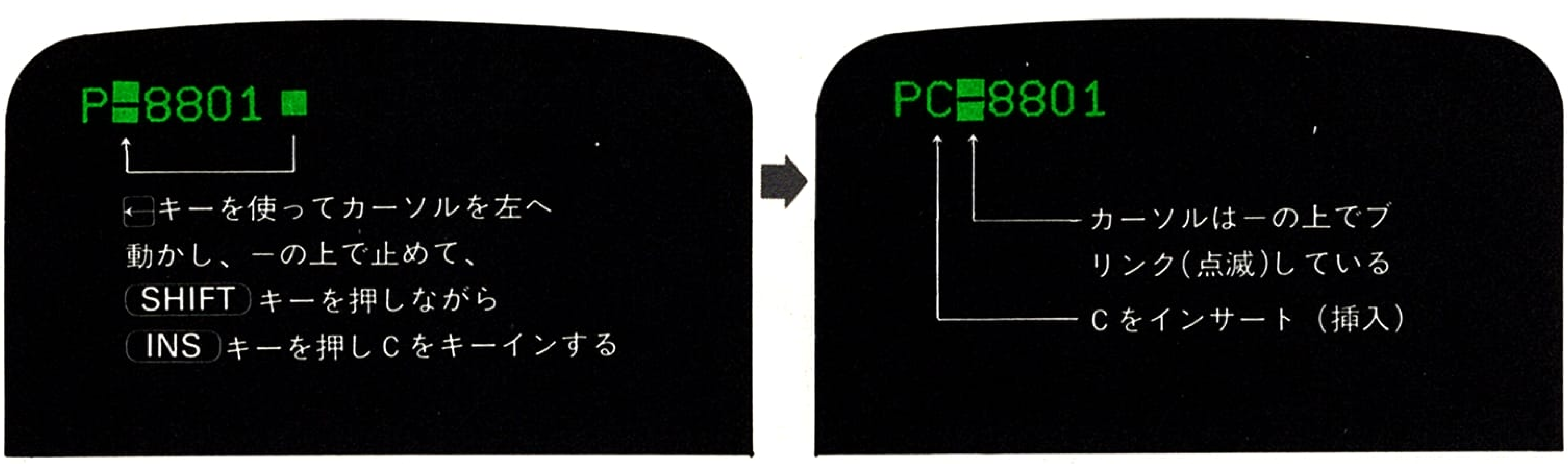


文字をインサート（挿入）するには

インサートモード

文字をインサート（挿入）したい場合には、その位置の右側の文字までカーソルを動かし、SHIFTキーを押しながらINSキーを押します。すると、インサートモードになり、何文字でも文字をインサートすることができます。カーソル移動キー（↑、←、→、↓）あるいは⏎キーを押すと、インサートモードから抜け出します。

たとえば、ＰＣ-８８０１をまちがってＰ-８８０１とキーインしたとしましょう。Ｃをインサートするには、←キーを使ってカーソルを左へ動かし、－の上で止めて、SHIFTキーを押しながらINSキーを押します。そして、Ｃをキーインします。



第2章

デモンストレーション

◆◆

2・1 デモプログラムのロード

付属のカセットテープを使用する場合

カセットテープレコーダがＰＣ−８８０１に接続されていますね。まず付属のカセットテープのＡ面がかかるように、カセットテープをカセットテープレコーダにセットします。

なお、カセットテープレコーダのＶＯＬＵＭＥ(ボリューム、音量)は中位に合わせ、ＴＯＮＥ（トーン、音質）は高目に合わせておきます。

次に、キーボードの CLR キーを押して、画面をクリアします。

ロード
ＬＯＡＤコマンド

カセットテープに録音されているプログラムをＰＣ−８８０１のメモリにロード（読み込み）するには、ＬＯＡＤ（ロード）コマンドを使って、次のようにキーインします。なお、ＬＯＡＤは小文字のｌｏａｄでもかまいません。

LOAD△"CAS:プログラム名" ⏎ ←―― カセットテープのプログラムをロード（読み込む）

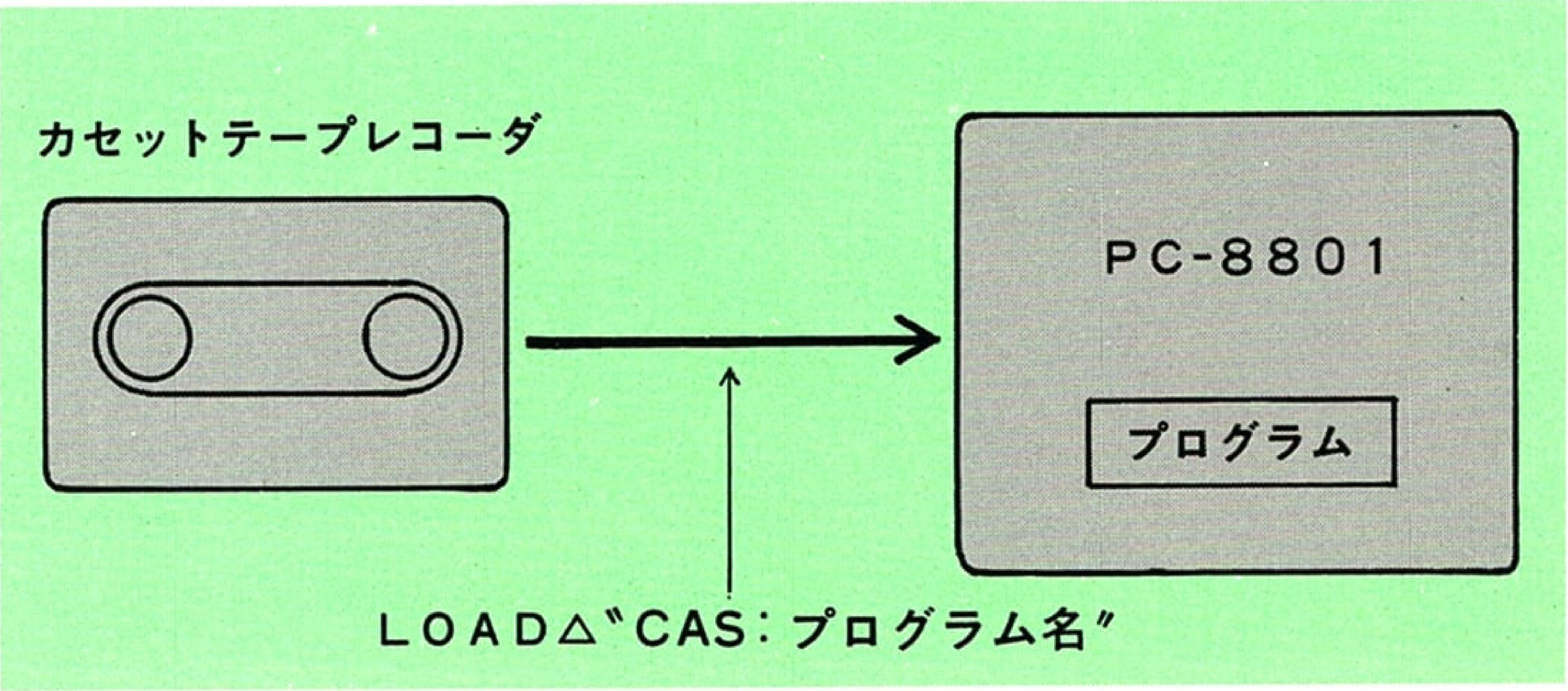
プログラム名 → プログラムの名前
△ → スペースを１つあける

ここで、ＣＡＳは、ｃａｓｓｅｔｔｅ ｔａｐｅ（カセットテープ）の最初の３文字だと考えましょう。小文字のｃａｓでもいいです。また

プログラム
プログラム名

プログラムとは、パソコンを動かすための命令書のようなものです。そして、プログラムの名前をプログラム名といいます。命令書とは何でしょうか。わかりませんね。とにかく、先へ進みましょう。

ＬＯＡＤコマンドによって、カセットテープに録音されているプログラムが、ＰＣ−８８０１のメモリにロード（読み込み）されます。

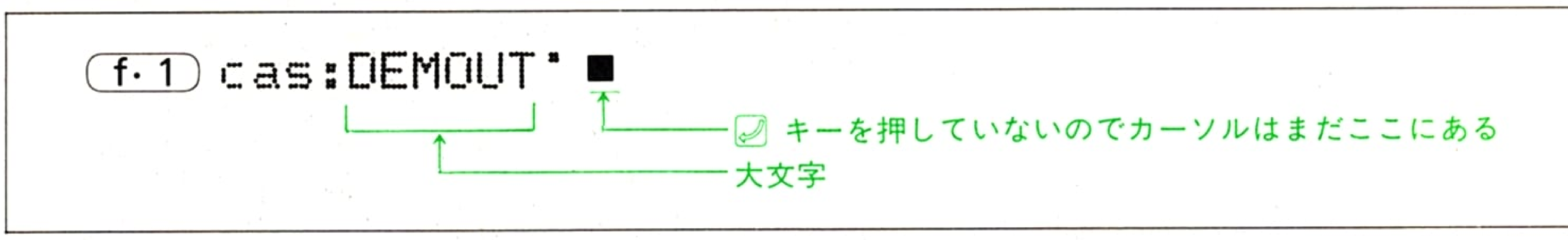


なお、ｌｏａｄ△"は f･1 キーに定義されていますので、f･1 キーを使いましょう。

付属のカセットテープに録音されているプログラムのプログラム名を録音順に示すと、次のとおりです。

❶ Ｑｕａｒｔｚ
❷ Ｃｐａｉｎｔ
❸ ＤＥＭＯ
❹ ＤＥＭＯＵＴ

これらのプログラムのうち、プログラムＤＥＭＯＵＴをロードしてみましょう。次のようにキーインします。



ここで、カセットテープレコーダの再生（ＰＬＡＹ）ボタンを押し、つづいて⏎キーを押します。

すると、カセットテープが回り始めます。

ＤＥＭＯＵＴ以外のプログラムが見つかると、画面に次のようなＳｋｉｐ（スキップ）の表示をして、次のプログラムを読みに行きます。



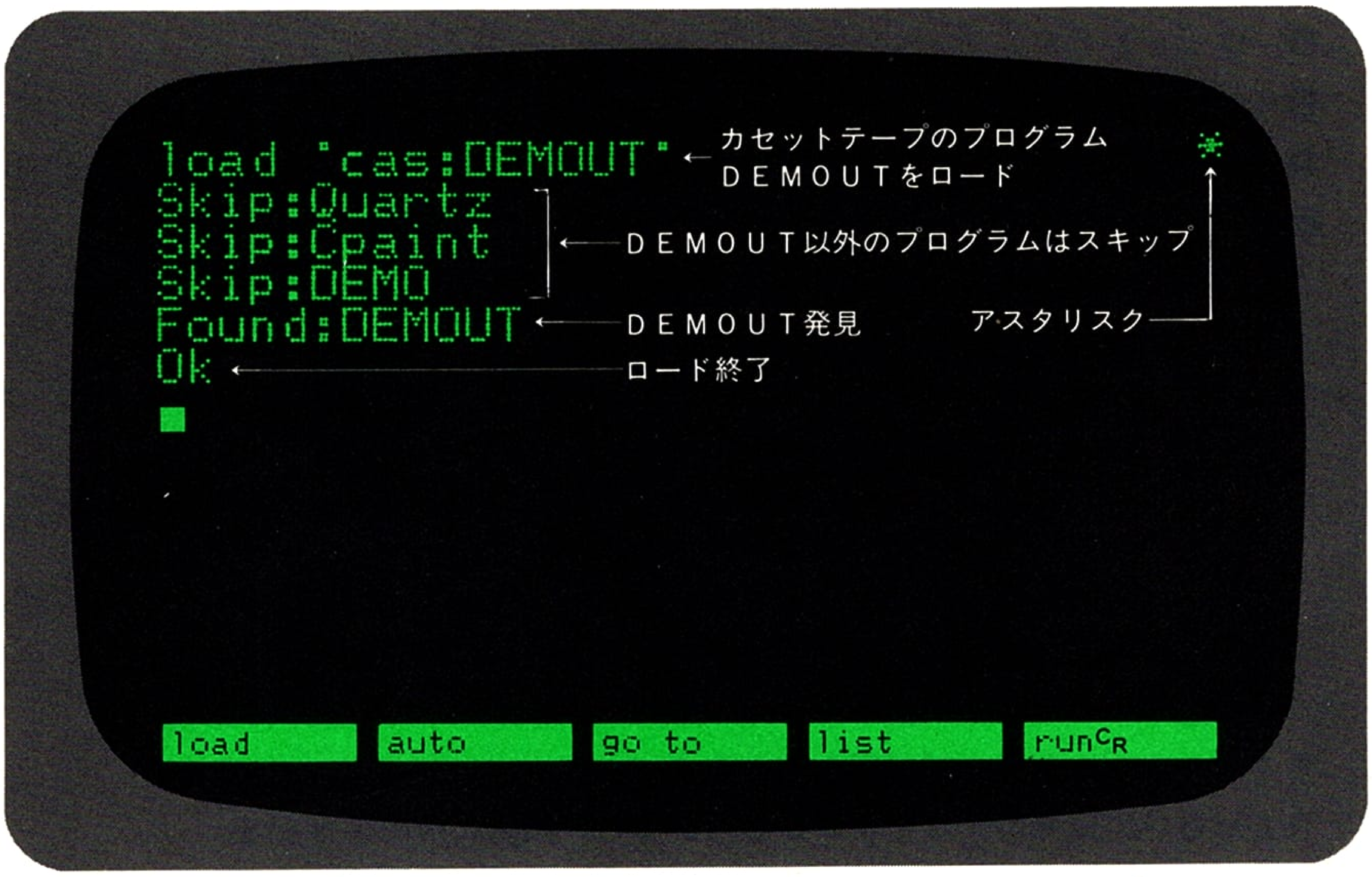
プログラムＤＥＭＯＵＴが見つかると、画面に次のように表示されます。



ロードが始まると、画面の右上スミに＊（アスタリスク）がブリンク（点滅）し、ロード中であることを示します。

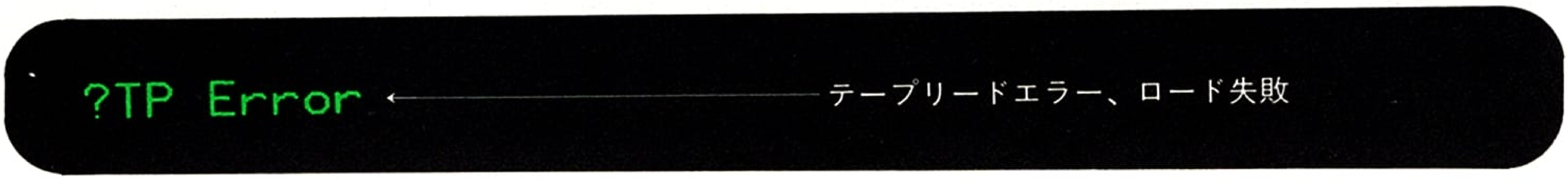
ロードが終わると、＊のブリンクが止まり、画面のＦｏｕｎｄの下にＯｋが表示されます。リモートが効いていれば、カセットテープは自動的に止まります。しかし、接続ケーブルの黒いプラグがカセットテープレコーダのリモート端子に接続してなかった場合とか、リモートの効かないテープレコーダの場合、テープレコーダのスタートやストップは自分で行う必要があります。リモートが効いていなければ、カセットテープは回り続けますので、カセットテープレコーダのストップ（ＳＴＯＰ）ボタンを押してカセットテープを止めておきます。

以上の手順を画面で見ると、次のようになります。

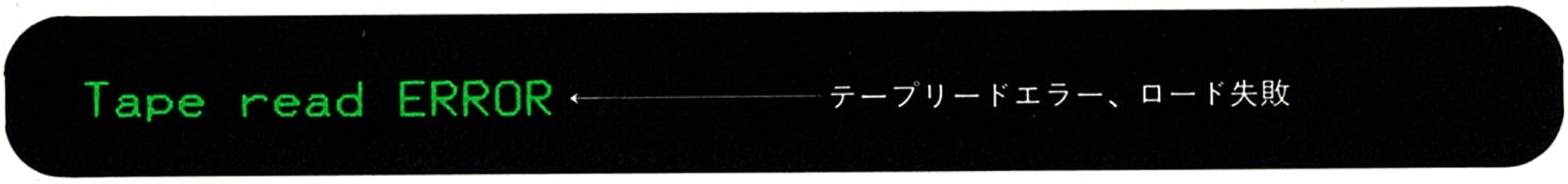


カセットテープからのプログラムのロードに失敗すると、ピーとブザーの音がして、エラーメッセージが表示されます。

ディスクを使わないときは、次のエラーメッセージが表示されます。



ディスク使用の場合のエラーメッセージは、次のようになります。



?TP Error
Tape read ERROR

「?TP Error」あるいは「Tape read ERROR」は、テープリードエラーのことで、カセットテープからのロード時にエラーが発生したことを意味します。

テープリードエラーの場合には、まず、カセットテープを巻き戻します。次に、VOLUME（音量）あるいはTONE（音質）を上げたり下げたり調整します。そして、もう一度LOADし直してください。

PC-8801とカセットテープレコーダがうまく接続されているのであれば、LOADが成功するまでこの手順を繰り返します。

カセットテープからのプログラムのロードがすんだら、カセットテープを巻き戻しておきましょう。

なお、カセットテープレコーダのリモート端子に接続ケーブルの黒いプラグが接続されている場合、カセットテープが巻き戻せないなど

うまく回り始めないことがあります。この場合は、次のようにキーインします。

```
MOTOR ⏎ ←—— カセットテープレコーダのモータのON、OFF
```

MOTOR文

このMOTOR（モータ）文を使えば、カセットテープレコーダのモータがONの状態であればOFFに、OFFの状態であればONになります。なお、MOTORは小文字のmotorでもかまいません。

カセットテープを巻き戻す必要がありますから、カセットテープレコーダのリモート端子には、接続ケーブルの黒いプラグを接続しないほうがよいでしょう。

システムディスクを使用する場合

ディスクユニット（フロッピィディスクユニットあるいはミニフロッピィディスクユニット）がPC-8801に接続されていますね。電源をONにし、システムディスクをディスクユニットのドライブ1にセットします。次に、リセットボタンを押します。画面に「How many files（0－15）？」のメッセージが表示されたら、⏎キーを押します。

FILESコマンド

ここで、FILES（ファイルズ）コマンドを使って、システムディスクに入っているファイルの名前を画面に表示してみましょう。次のようにキーインします。なお、FILESは、小文字のfilesでもかまいません。

```
FILES ⏎ ←—— ファイル名を画面に表示
```

画面に、次のように表示されます。ファイル名の中に、プログラムファイルDEMOUTの名前がありますね。

```
files
backup.n88 1    format.n88 1    setinf.n88 1    xfiles.n88 1    Quartz.     1
Cpaint.    1    DEMO  .    1    DEMOUT.    2
Ok
```

↑—— プログラムファイルDEMOUT

ファイル

プログラムファイル

データファイル

なお、ファイルとは、意味をもつ情報の集まりのことです。テープレコーダやディスクユニットなどの入出力を行う場合、ファイルが1つの単位となります。ファイルには、プログラムファイルとデータファイルの2つがあります。したがって、先ほどテープレコーダを使用する場合、単にプログラムと呼んでいましたが、正確にはプログラムファイルと呼ぶべきです。しかし、別にプログラムでもかまいません。とにかく、次に進みましょう。

次に、ファイルの名前をプリンタに出力してみましょう。プリンタがPC-8801に接続されていますね。プリンタの電源をONにし、SEL（セレクト）スイッチを押して、プリンタをSELECT（セレクト）状態にします。

LFILESコマンド

ファイル名をプリンタに出力するには、LFILES（エルファイルズ）コマンドを使います。次のようにキーインしてみましょう。

なお、LFILESも、小文字でもかまいません。

```
LFILES ⏎  ←――― ファイル名をプリンタに出力
```

プリンタには、次のように出力されます。DEMOUTがありますね。

```
backup.n88 1    format.n88 1    setinf.n88 1    xfiles.n88 1    Quartz.    1
Cpaint.    1    DEMO   .   1    DEMOUT.    2
                                └─┘←― プログラムファイルDEMOUT
```

LOADコマンド

ドライブ1のディスクに入っているプログラムをロードするには、LOADコマンドを使って、次のようにキーインします。

なお、LOADは小文字のloadでもかまいません。

```
LOAD△"プログラム名" ⏎  ←――― ドライブ1のプログラムをロード
    ↑
    └――――――――――――――――― スペースを1つあける
```

LOADコマンドによって、ディスクに入っているプログラムがPC-8801のメモリにロードされます。

なお、(f・1)キーにload△"が定義されていますね。

それでは、LOADコマンドを使って、DEMOUTをシステムディスクからロードしましょう。

次のようにキーインします。

```
(f・1) :DEMOUT" ⏎  ←――― DEMOUTをロードせよ
        └────┘
          ↑
          └――――――――――― 大文字
```

⏎キーを押すと、すぐにシステムディスクからロードされます。ロードが終わると、画面のloadの下にOkが表示されます。画面表示は次のようになります。

```
load "DEMOUT"
Ok
```

2·2　デモプログラムの実行

プログラムDEMOUTを実行しよう

RUNコマンド

プログラムの実行には、RUN(ラン)コマンドを使います。次のようにします。なお、RUNは小文字のrunでもかまいません。

RUN ⏎ ←プログラムの実行

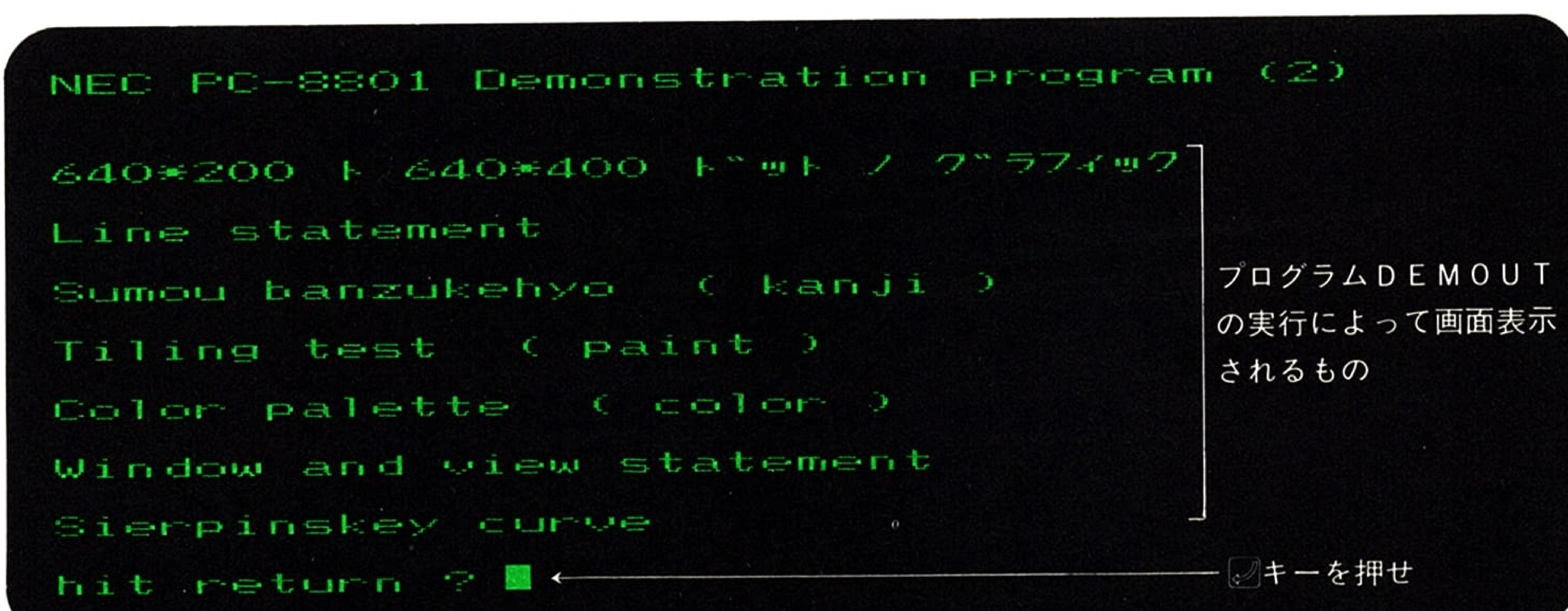
しかし、run $^{C}_{R}$ が f·5 キーに定義されているので、 f·5 キーを使いましょう。

付属のカセットテープあるいはシステムディスクから、プログラムDEMOUTがPC-8801のメモリにロードされていますね。

それでは、 f·5 キーを押しましょう。 ⏎キーを押す必要はありません。 f·5 キーを押すだけで、ただちにDEMOUTを実行します。

f·5 ← f·5 キーを押す。⏎キーを押す必要はありません

画面は、次のようになります。



⏎キーを押します。

画面にグラフが表示されましたね。あなたの持っているディスプレイがカラーであれば、きれいな色が楽しめます。

プログラムDEMOUTの実行につれ、いくつかのメッセージが画面に表示されます。次のようなものです。

- 専用高解像度ディスプレイを接続していますか？
  ハイ・・・〔y〕　イイエ・・・〔n〕？
- hit return？

●カンジＲＯＭボードが内蔵されていますか？
　ハイ・・・〔ｙ〕　イイエ・・・〔ｎ〕？

「ｈｉｔ　ｒｅｔｕｒｎ？」のメッセージに対しては、⏎キーを押しましょう。その他の質問には、「ハイ」であればYと⏎キーを押し、「イイエ　であればNと⏎キーを押します。

ＰＣ-８８０１のデモンストレーションは、いかがですか？

実にきれいですね。

プログラムＤＥＭＯＵＴの実行が終わると、画面左上にＯｋが表示されます。

ここで、f･5キーを押して、もう一度ＤＥＭＯＵＴを実行しましょう。

プログラムの実行をストップさせたいとき

STOPキー

プログラムの実行を途中でストップ（停止）させるには、キーボード左上スミのこげ茶色のSTOP（ストップ）キーを押します。やってみましょう。

そして、f･5キーを押して、再度、ＤＥＭＯＵＴを実行します。

画面の表示を一時的にストップさせたいとき

CTRLキー

キーボードの左側にこげ茶色のCAPSキーがありましたね。その隣りにCTRL（コントロール）キーがあります。

画面の表示を一時ストップ（停止）させるには、CTRLキーを押しながら、Sのキーを押します。

もう一度CTRLキーを押しながらSキーを押せば、画面の表示が再開されます。やってみましょう。

画面をプリンタにコピーしてみよう

COPYキー

キーボード左上にCOPY（コピー）キーがありますね。

画面上に表示されている文字や図形をそっくりプリンタに出力したいときは、画面の表示が一時ストップしている状態で、COPYキーを押します。

ここで、画面の表示が一時ストップしている状態とは、次のような場合をいいます。

●画面にメッセージ（たとえば「ｈｉｔ　ｒｅｔｕｒｎ？」）が表示れ、ＰＣ-８８０１があなたのキーインを待っている状態
●STOPキーを押して、プログラムの実行をストップさせた状態
●CTRLキーを押しながらSキーを押して、画面の表示を一時ストップさせた状態

こうした状態で、もしプリンタに出力させたくない文字などがあれば、カーソル移動キー（↑、←、→、↓）を使ってカーソルを動かし、DEL キーや ＿＿ キーによって不要文字を削除します。その上で COPY キーを押します。

すると、画面の内容がそっくりプリンタ用紙にコピーされます。このコピーのことをハードコピーといい、画面上に表示されているものをソフトコピーといいます。

ハードコピー
ソフトコピー

さあ、プログラムＤＥＭＯＵＴのハードコピーをとってみましょう。

ただし、あなたのプリンタが、ＰＣ-８８０１のＣＯＰＹ機能が実現できるプリンタ（たとえば、ＰＣ-８８２１やＰＣ-８８２２）でないと、ハードコピーはとれません。また、ハードコピーはすべて白黒です。

次に、ＤＥＭＯＵＴのハードコピー例を３つ示しておきます。

❶　６４０＊２００　ドット　ノ　グラフ

「６４０＊２００ドット」については、あとで説明します。

❷　寛政５年大相模秋場所番付

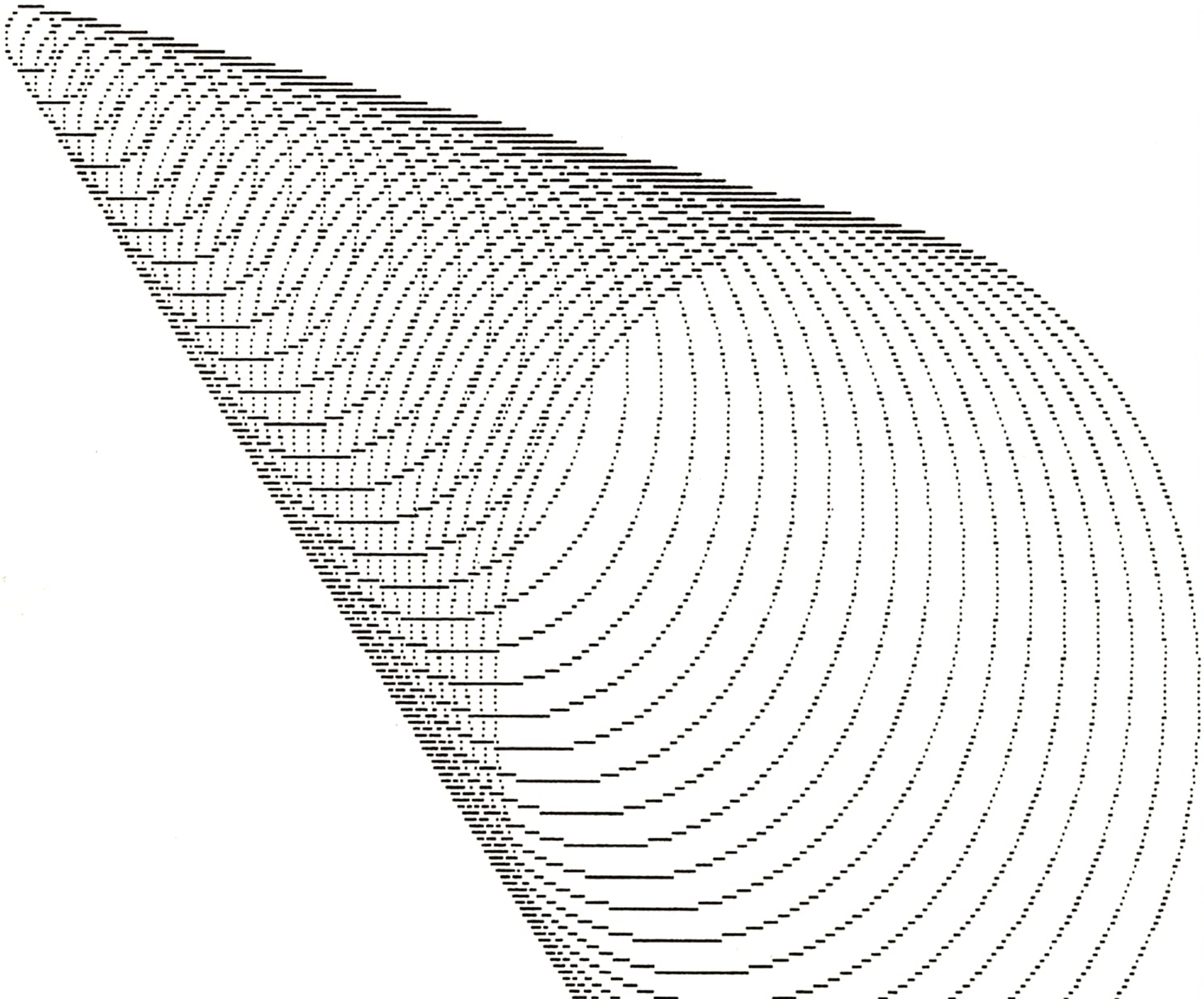
この相模番付のハードコピーをとるためには、まずＰＣ-８８０１本体内にＰＣ-８８０１－０１（ＰＣ-８８０１用漢字ＲＯＭボード）が実装されていなければなりません。そして、プリンタが漢字プリンタ（たとえば、ＰＣ-８８２２）か漢字ＲＯＭボードを実装したプリンタ（たとえば、ＰＣ-８８２１にＰＣ-８８２１－０２〈ＰＣ-８８２１用漢字ＲＯＭボード〉を追加したもの）でなければなりません。

❸　ｗｉｎｄｏｗ　ａｎｄ　ｖｉｅｗ　ｓｔａｔｅｍｅｎｔ

ｗｉｎｄｏｗ文やｖｉｅｗ文については、本書の範囲をこえますが、デモストレーションだけでもお楽しみください。

さて、デモンストレーションプログラムには、ＤＥＭＯＵＴのほかに、Ｑｕａｒｔｚ、Ｃｐａｉｎｔ、ＤＥＭＯがありましたね。これらのプログラムもロードして、実行してみましょう。

640*200 ﾄﾞｯﾄ ﾉ ｸﾞﾗﾌｨｯｸ



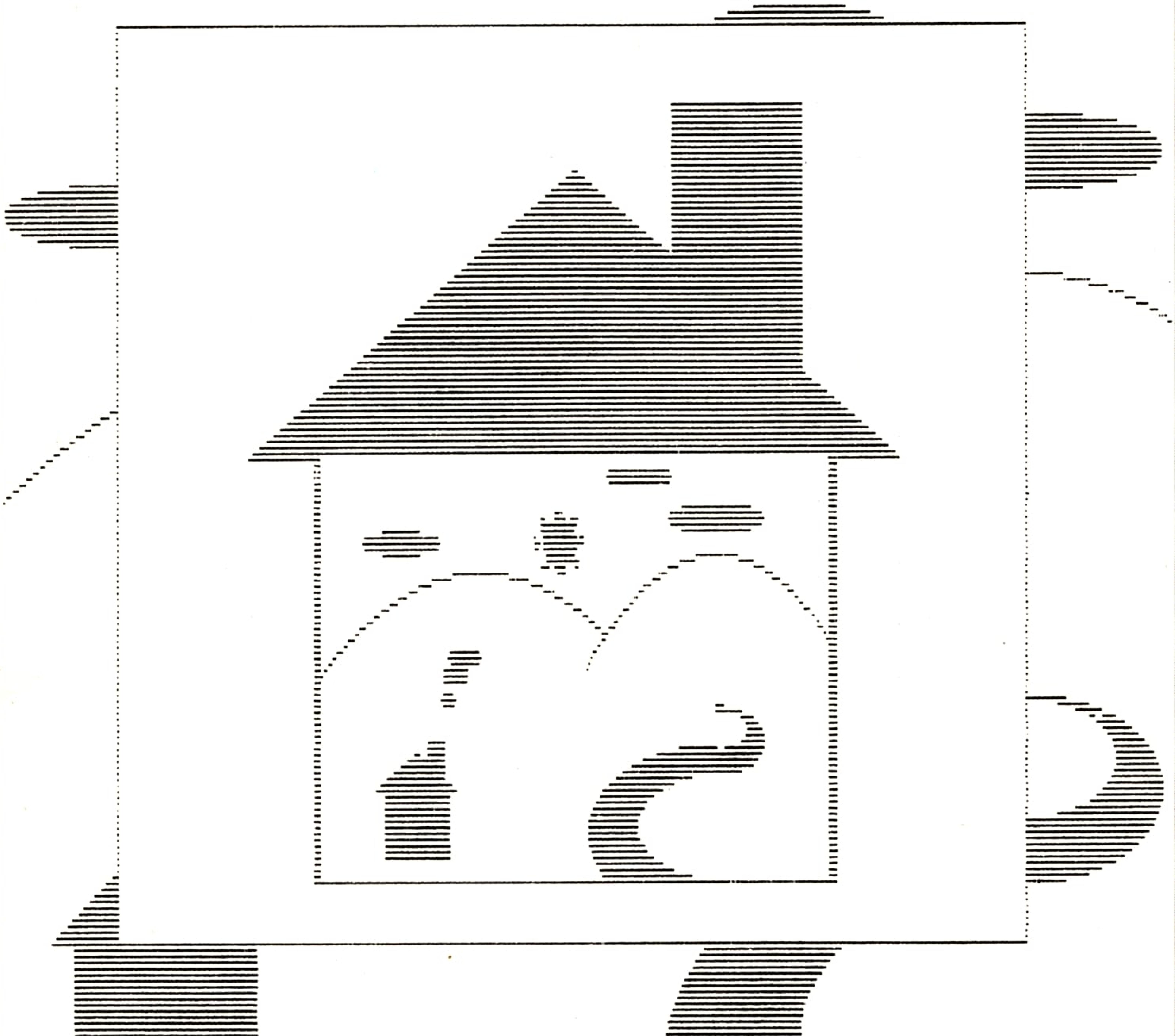
寛政五年大相撲秋場所番付

西

前七　達ヶ関　森右エ門
前六　出羽ノ海　運右エ門
前五　関ノ戸　八郎治
前四　沖津岡　梶右エ門
前三　越ノ戸　浜右エ門
前二　綿木塚　右エ門
前一　甲斐嶽　岨右エ門
小結　鶴渡　岡右エ門
関脇　雷電　為右エ門
大関　谷風　梶之助

東

大関　小野川　才助
関脇　陣幕　島之助
小結　九紋龍　清吉
前一　玉垣　額之助
前二　勢見山　兵右エ門
前三　和田ヶ原　甚四郎
前四　門司関　勘太夫
前五　永尾山　大五郎
前六　伊勢浜　荻右エ門
前七　八ッケ峰　住右エ門



第3章

電卓がわりに使うには

3

3・1　四則計算をしてみよう

四則計算とは

たし算（加法）、ひき算（減法）、かけ算（乗法）、わり算(除法)の４つの計算を四則計算といいます。すなわち、四則計算とは、和・差・積・商を求める計算です。

ここで、１＋２－３×４÷５の計算をしてみましょう。

電卓の場合は

電卓の場合、計算式につづいて[=]（イコール）キーを押せば、計算結果が表示されます。

パソコンの場合は

キーボードに[+]や[−]のキーはありますが[×]や[÷]のキーはありません。かけ算の記号としては[*]（アスタリスク）キーを使い、わり算の記号には[/]（スラッシュ）キーを使います。これらのキーは、キーボード右側のテンキーのところにありますね。

しかし、パソコンの場合、計算式につづいて[=]キーを押しても、計算結果を画面に表示しません。

PRINT文

計算結果を画面に表示させるには、ＰＲＩＮＴ（プリント）文を使って、次のようにキーインします。なお、ＰＲＩＮＴは小文字のｐｒｉｎｔでもかまいません。

PRINT △ 計算式 ⏎ ← 計算結果を画面に表示
↑ スペースを１つあける

しかし、ｐｒｉｎｔ△はｆ・８に定義されていましたね。(SHIFT)キーを押しながら(f·3)キーを押せばよいのです。

それでは、１＋２－３＊４÷５の計算をしてみましょう。次のようにキーインします。なお、(SHIFT)キーを押しながら(f·3)キーを押すことを(SHIFT)＋(f·3)と表示することにします。

(SHIFT)＋(f·3) 1+2−3*4/5 ⏎

- / ← パソコンで使うわり算の記号（スラッシュ）
- * ← パソコンで使うかけ算の記号（アスタリスク）
- (SHIFT)＋(f·3) ← (SHIFT)キーを押しながら(f·3)キーを押す

画面には、次ページのように表示されます。



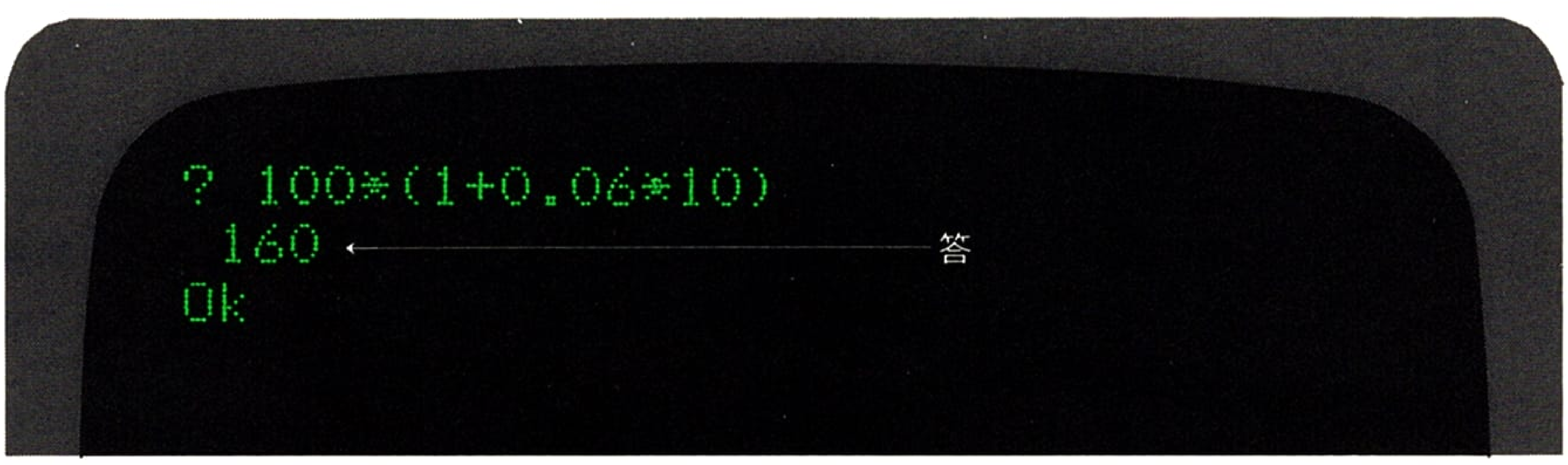
ｐｒｉｎｔのかわりに、？（クエッションマーク）を使うことができます。？を使って、100×（１＋0.06×10）の計算をしてみましょう。？を画面に表示するには、(SHIFT)キーを押しながら[?]キーを押します。 ?

```
?△100*(1+0.06*10)⏎
```

？の利用
カッコの中を先に計算
スペースをあけたほうが見やすい

四則の混じった計算では、かけ算・わり算を先にし、たし算・ひき算はあとにします。しかしカッコがある場合には、カッコの中から先に計算します。

実行結果は次のようになります。



ｐｒｉｎｔを使うか？を使うかは好みの問題です。

ｐｒｉｎｔを使う場合には、(SHIFT)キーを押しながら(f·3)キーを押します。他方、？を使う場合には、(SHIFT)キーを押しながら[?]キーを押します。手数は同じですね。しかし、(SHIFT)キーを押しながら(f·3)キーを押せば、「ｐｒｉｎｔ△」とｐｒｉｎｔのあとに自動的にスペースが１つあきます。他方、？のほうは、スペースを１つあけるためには、(　　　)キーを１回押さなければなりません。両者のちがいは、スペースが１つあくかどうかのちがいです。もっとも、ｐｒｉｎｔと？では、どちらがわかりやすいか、あるいはどちらが見ばえがよいかなどの差もありますね。

あなたは、ｐｒｉｎｔと？とでは、どちらが好みに合いますか。好きなほうを使ってください。

3・2　複利計算をしてみよう

100万円預けると10年後にはいくらになるか

元金Ｐを年利率Ｒ％の複利でＮ年間預けた場合の元利合計Ｓの計算式は、次のとおりです。

$$S = P \times \left(1 + \frac{R}{100}\right)^{N}$$

なお、年利率をＲ％（たとえば６％）のように％表示することにし、元利合計Ｓを計算するにあたって、Ｒを100で割って小数点以下の数値に直しておきます。

また、パソコンでは、累乗（ベキ乗）の計算には、∧（ヤマガタ）を使います。

それでは、元金100万円を年利率6％の複利で10年間預けた場合の元利合計を計算してみましょう。

次のようにキーインします。

```
SHIFT + f･3  100*(1+6/100)^10 ↵
```

^ …… パソコンで使う累乗の記号（ヤマガタ）

(1+6/100) …… カッコの中から計算する

SHIFT + f･3 …… SHIFT キーを押しながら f･3 キーを押す

四則の混合計算の中に累乗（ベキ乗）がはいっている場合は、累乗の計算（ベキ算）を先にします。しかし、カッコがある場合には、カッコの中から先に計算します。

画面は、次のようになるはずですが、いかがでしょうか？

```
print 100*(1+6/100)^10
 179.085
Ok
```

179.085 ← 元利合計

元金100万円を年利率6％の複利で10年間預けた場合の元利合計は、179万850円ということになります。

さて、ここで、年利率を６％から７％に変えてみましょう。元利合計はいくらになるでしょうか。あなたはどうしますか？

とりあえず、次のようにキーインしてみましょう。

SHIFT + f·3 100*(1+7/100)^10 ⏎

しかし、これでは、最初からキーインしなければならず少々めんどうです。次のようにしてみましょう。

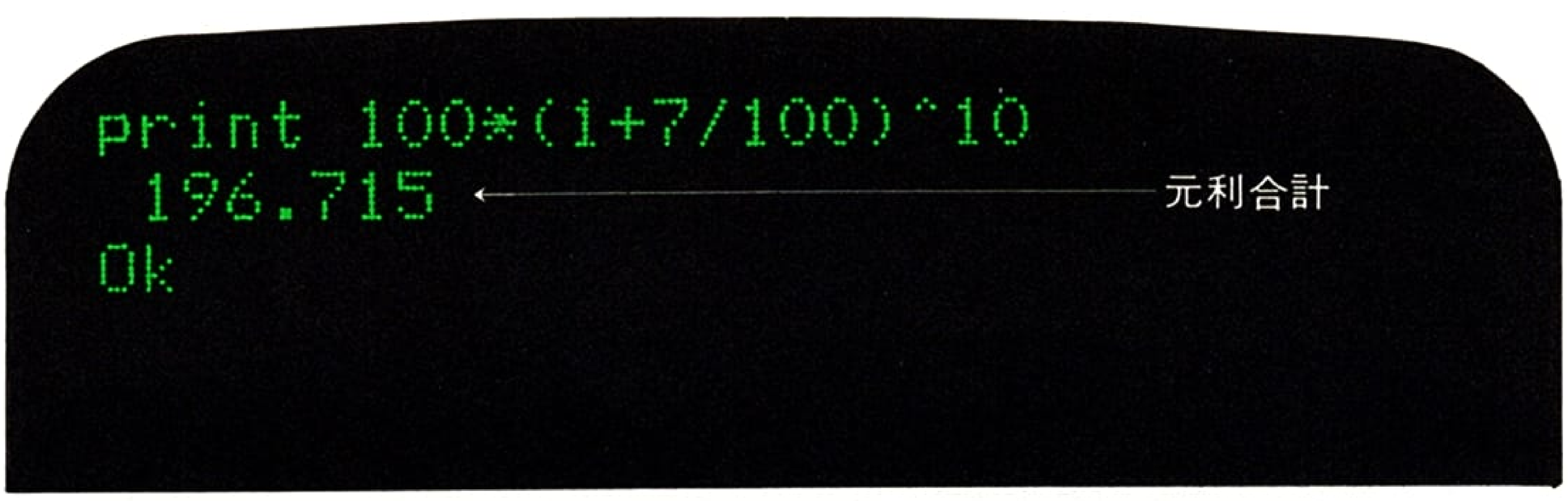
まず、↑、→を使ってカーソルを動かし、年利率 6 /100の 6 の上で止めます。ここで 7 をキーインします。そして、⏎キーを押します。なお、カーソルを年数10のうしろに動かしてから⏎キーを押すという必要はありません。

```
print 100*(1+6/100)^10
 179.085
Ok
```

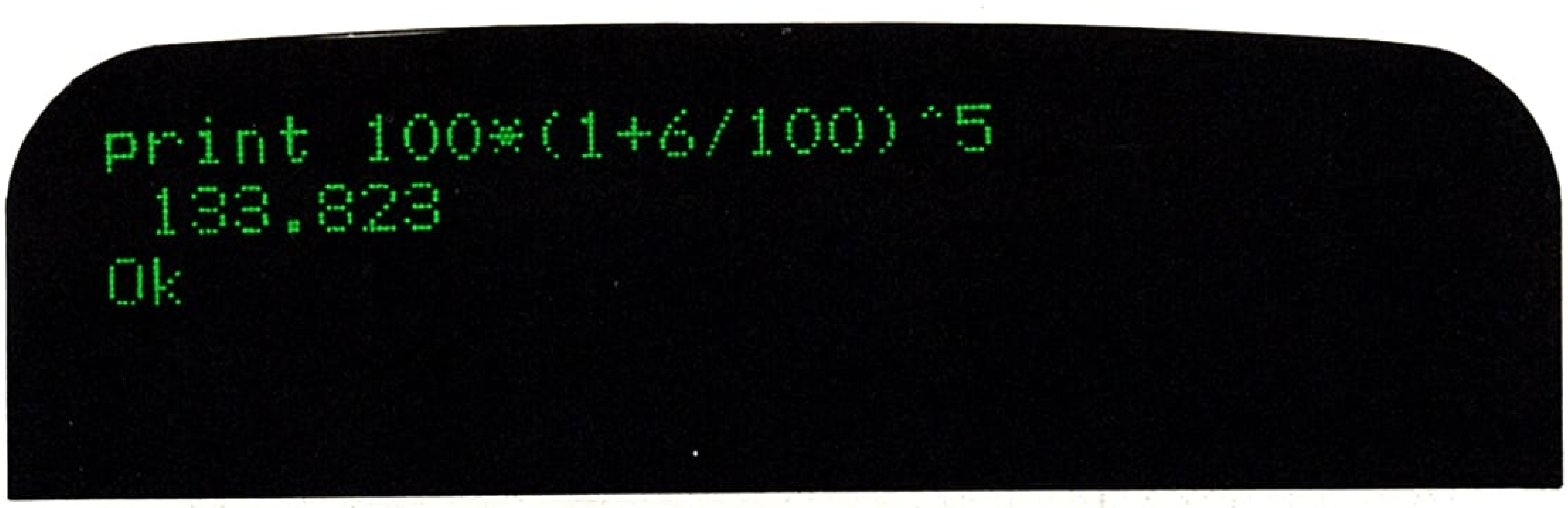
① ↑、→キーを使ってカーソルを動かし、6の上で止める
② 7をキーインする
③ 6が7にかわり、カーソルは／の上でブリンク(点滅)する

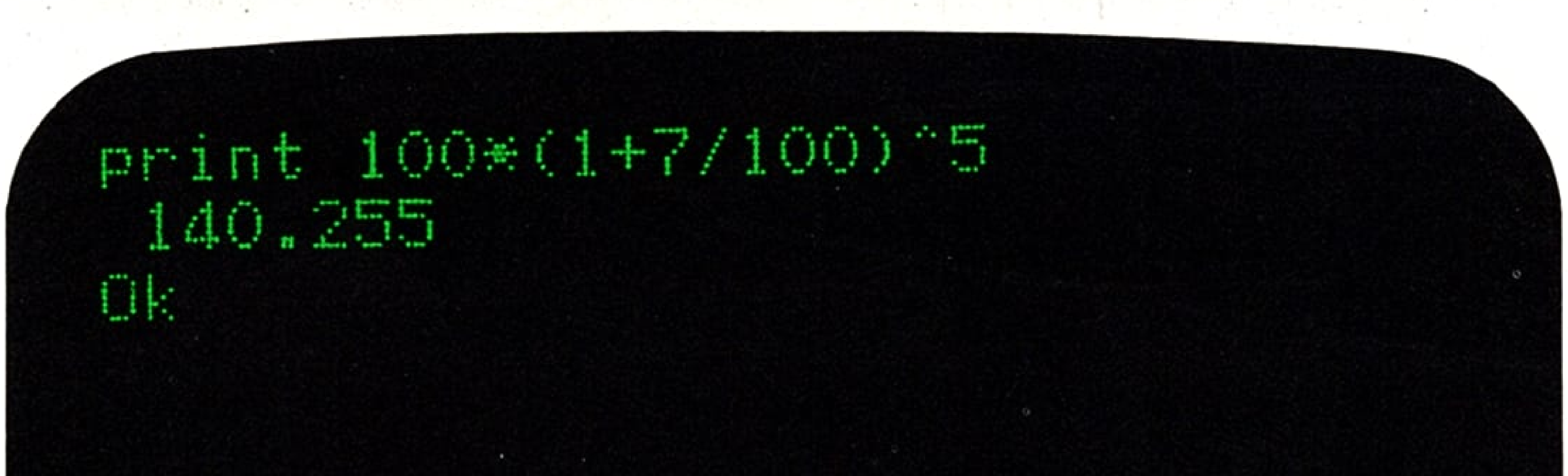
ここで⏎キーを押す

画面は次のようになります。



カーソル移動キーを使ってカーソルを動かし、元金、年利率、年数を変えて、元利合計を繰り返し計算してみましょう。





3·3　平方根の計算もできます

平方根の値を求めてみよう

SQR関数

ある数aの平方根 $\sqrt{a}$（ルートa）の値を求めるには、SQR（スクェアルート）関数を使って、次のようにキーインします。なお、SQRは小文字のsqrでもかまいません。

```
PRINT△SQR(a) ⏎
```

（$\sqrt{a}$ として、数値あるいは数式をキーイン／△：スペースを1つあける）

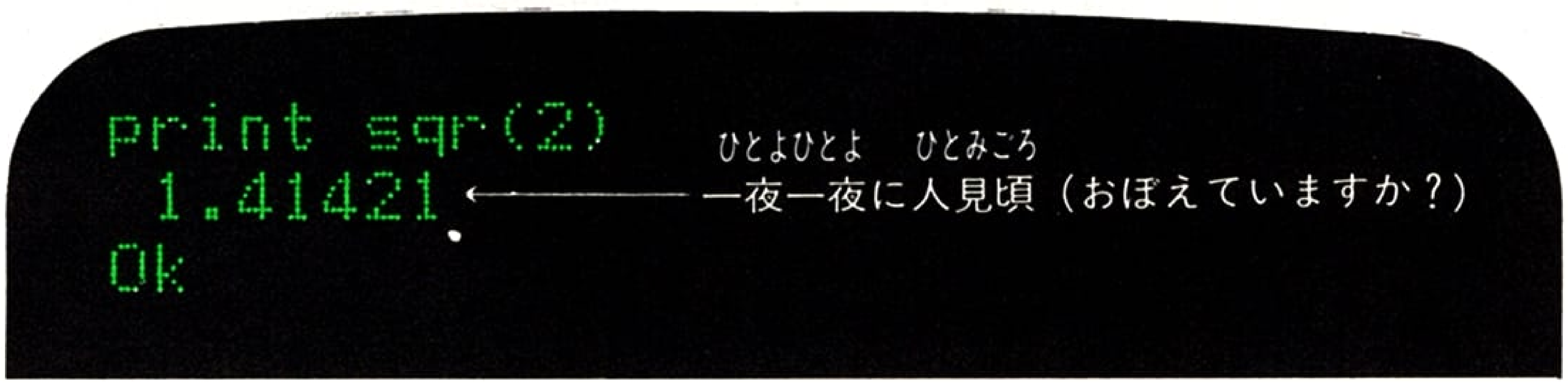
ただし、aの値が正または0でなければ、エラーメッセージが表示されます。

なお、関数とは、ある数に対してあるきまった計算をするもので、その計算結果がえられます。たとえば、SQR関数は、ある数aに対して平方根 $\sqrt{a}$ の計算をするもので、$\sqrt{a}$ の値がえられます。

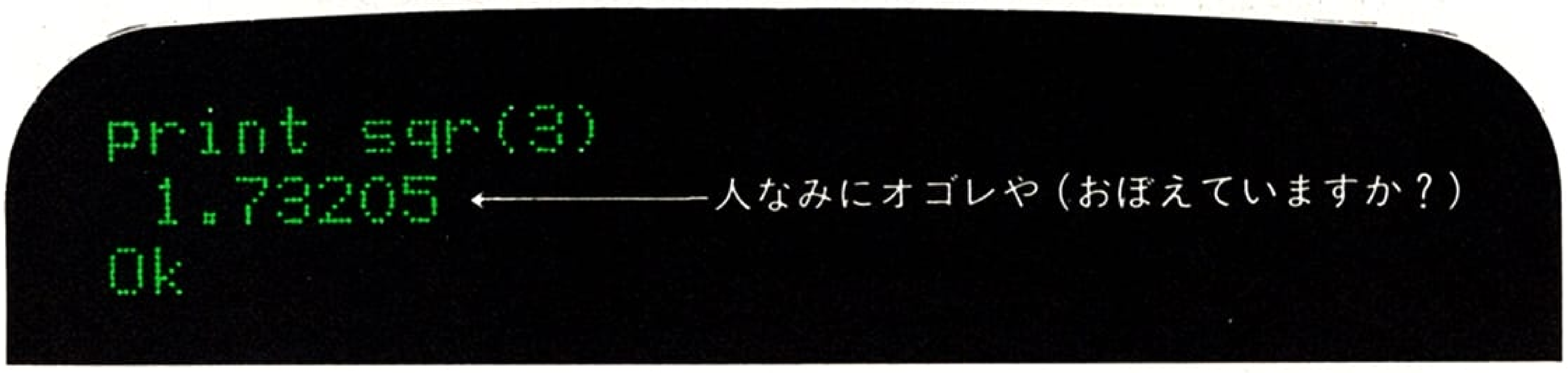
それでは、$\sqrt{2}$ の値を求めてみましょう。次のようにキーインしてください。

```
SHIFT + f·3   sqr(2) ⏎
```

画面に、次のように表示されます。



$\sqrt{3}$ の値を求めてみましょう。カーソル移動キーを使ってカーソルを動かし、2を3にかえ、⏎キーを押します。次のようになります。



同じようにして、$\sqrt{5}$、$\sqrt{6}$、$\sqrt{7}$、……などの値を計算してみましょう。

$\sqrt{a}$ を計算する場合、aの値は正または0でなければなりません。しかし、ためしに、SQR（−2）とキーインしてみましょう。次のようにエラーメッセージが表示されます。

ディスクを使わない場合は、次のようになります。

```
print sqr(-2)
?FC Error          ← エラーメッセージ
Ok
```

ディスクを使う場合は、次のようになります。

```
print sqr(-2)
Illegal function call   ← エラーメッセージ
Ok
```

?FC Error

Illegal function call

「?FC　Error」あるいは「Illegal　function　call」は、「関数における機能の呼び方が誤っている」というPC-8801からあなたへのエラーメッセージです。「関数における機能の呼び方」とは何でしょうか。わからなくても、気にせず、先に進みましょう。ここでは、$\sqrt{a}$のaの値が正または0でなければならないことだけ確かめておけばよいのです。

2次方程式の解の値を計算してみよう

2次方程式$ax^2+bx+c=0$の解は、

$$X=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

です。この式を2次方程式の解の公式といいます。

忘れてしまいましたか？　しかしここではそんなことはどうでもいいのです。とにかく実際にやってみましょう。2次方程式$2x^2+4x-1=0$の解の値はどうなるでしょうか。a＝2、b＝4、c＝－1としてパソコンに計算させましょう。

```
            ┌─＋を－にかえ⏎キーを押せばもう1つの解がえられる
            ↓
print (-4+sqr(4^2-4*2*(-1)))/(2*2)
 .224745   ←──┐
Ok             ├── 2つの解
print (-4-sqr(4^2-4*2*(-1)))/(2*2)
-2.22475   ←──┘
Ok
```

なおパソコンでは、カッコは｛　｝や［　］を使わずに、すべて（　）を使います。また、（と）の数が同じかどうかチェックしてください。

倍精度関数機能を使ってみよう

単精度型
倍精度型

整数や分数（小数）、平方根や円周率π（パイ）などをあわせて実数といいましたね。ＰＣ-８８０１では、実数は単精度型と倍精度型に分けられます。単精度型の場合、有効桁は７桁の精度であり、６桁以下で表示されます。また、倍精度型の場合、有効桁は16桁の精度であり、16桁以下で表示されます。

ＳＱＲ関数など関数は、数が整数や単精度のとき、関数の値が単精度となります。たとえば、ＳＱＲ（２）の値は、1.41421でしたね。

倍精度関数機能

これに対し、ディスクを使用する場合にかぎって、数が倍精度のとき、関数の値は倍精度となります。この機能を倍精度関数機能といいます。

倍精度関数機能を使って$\sqrt{2}$ の値を計算するには、２のうしろに＃（シャープ）をつけて、次のようにキーインします。

SHIFT ＋ f·3 sqr(2#) ↵ ← 倍精度による$\sqrt{2}$ の計算

それでは、倍精度関数機能を使って、$\sqrt{2}$、$\sqrt{3}$、$\sqrt{5}$、$\sqrt{6}$、$\sqrt{7}$ の値を計算してみましょう。次のようになります。

```
print sqr(2#)
 1.414213562373095
Ok
print sqr(3#)
 1.732050807568877
Ok
print sqr(5#)
 2.23606797749979
Ok
print sqr(6#)
 2.449489742783178
Ok
print sqr(7#)
 2.6457513110645091
Ok
```

1.414213562373095 ← 一夜一夜（ひとよひとよ）に人見頃（ひとみごろ）
1.732050807568877 ← 人なみにオゴレや
2.23606797749979 ← 富士山麓（ふじさんろく）オーム鳴く
2.449489742783178 ← 似（に）よ、よくよく
2.6457513110645091 ← 菜（な）に虫いない
おぼえていますか

ディスクを使用しない場合、$\sqrt{2}$ の値は次のようになります。

ＳＱＲ（２）　＝1.41421
ＳＱＲ（２＃）＝1.41421

ディスクを使用する場合、$\sqrt{2}$ の値は次のようになります。

ＳＱＲ（２）　＝1.41421
ＳＱＲ（２＃）＝1.414213562373095

倍精度関数機能が使えるのは、ディスクを使用する場合のみです。

3·4　関数計算もできます

ＰＣ-８８０１によって、ＳＱＲ関数など多くの関数の値を計算することができます。ここでは、対数関数と三角関数の値を計算してみましょう。

対数関数の値を求めてみよう

ＬＯＧ関数

常用対数 $\log_{10}X$ の値を求めるには、ＬＯＧ(ロガリズム)関数を使って、次のようにキーインします。なお、ＬＯＧは小文字のｌｏｇでもかまいません。

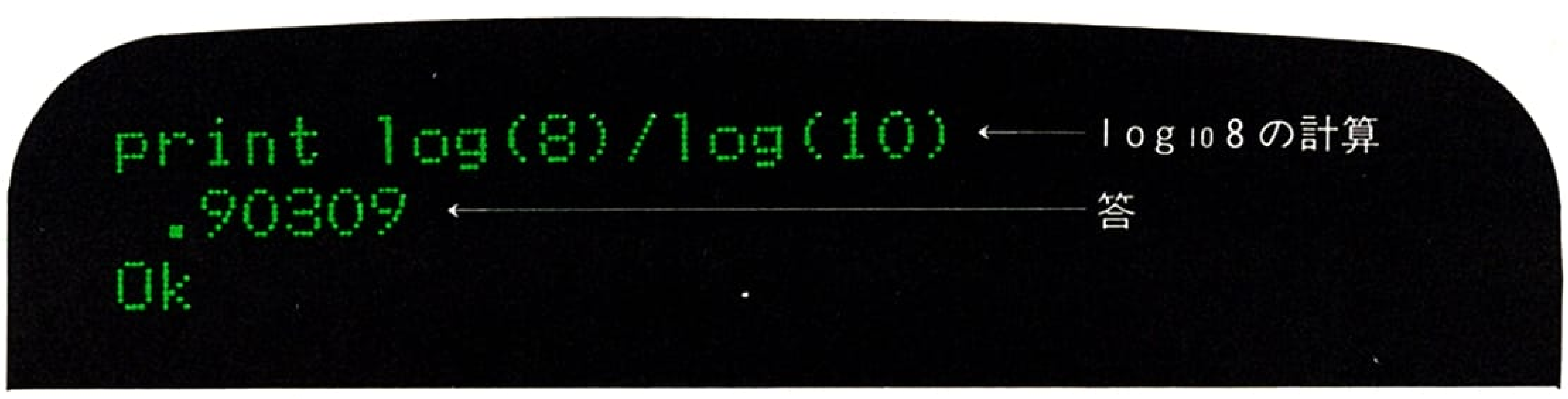
```
PRINT △ LOG(x)/LOG(10) ⏎
```

← ｘとして　数あるいは数式をキーイン

ただし、ＬＯＧ（Ｘ）は、ｅを底とした自然対数 $\log_e X$ です。また、Ｘ＞ｏです。

それでは $\log_{10}8$ の値を求めてみましょう。

次のようになります。



次に、ａを底とするｘの対数関数 $\log_a X$ の値を求めてみましょう。ＬＯＧ関数を使って、次のようにキーインします。

```
PRINT△ LOG(x)/LOG(a) ⏎
```

← ｘ、ａに　数あるいは数式をキーイン

ただし、ｘ＞ｏ、ａ＞ｏ、ａ≠１′です。

それでは、$\log_2 8$ の値を求めてみましょう。次のようになります。



三角関数の値を求めてみよう

ＳＩＮ関数

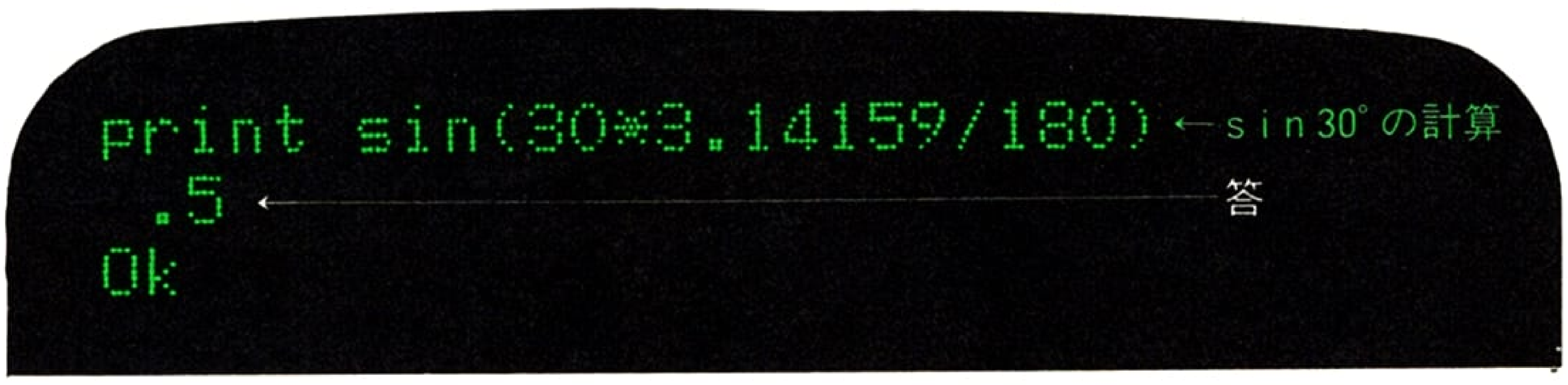
三角関数 sin d°（度数法)の値を求めるには、ＳＩＮ（サイン）関数を使って、次のようにキーインします。なお、ＳＩＮは小文字のｓｉｎでもかまいません。

```
PRINT△SIN(d*3.14159/180) ⏎
```

← dの単位は度

ただし、円周率 π =3.14159とします。

それでは、sin30°の値を求めてみましょう。次のようになります。

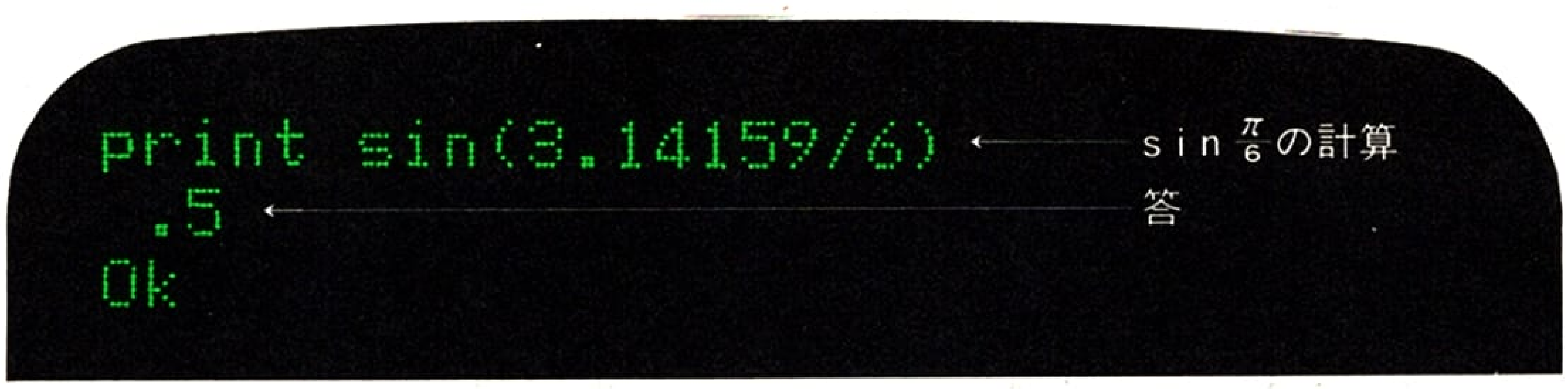


また、三角関数sin x（弧度法）の値を求めるには、SIN関数を使って、次のようにキーインします。ただし、xの単位はラジアン（π ラジアン=180°）です。

```
PRINT△SIN(x) ⏎
```

← xの単位はラジアン

それでは、sin $\frac{\pi}{6}$ の値を求めてみましょう。次のようになります。



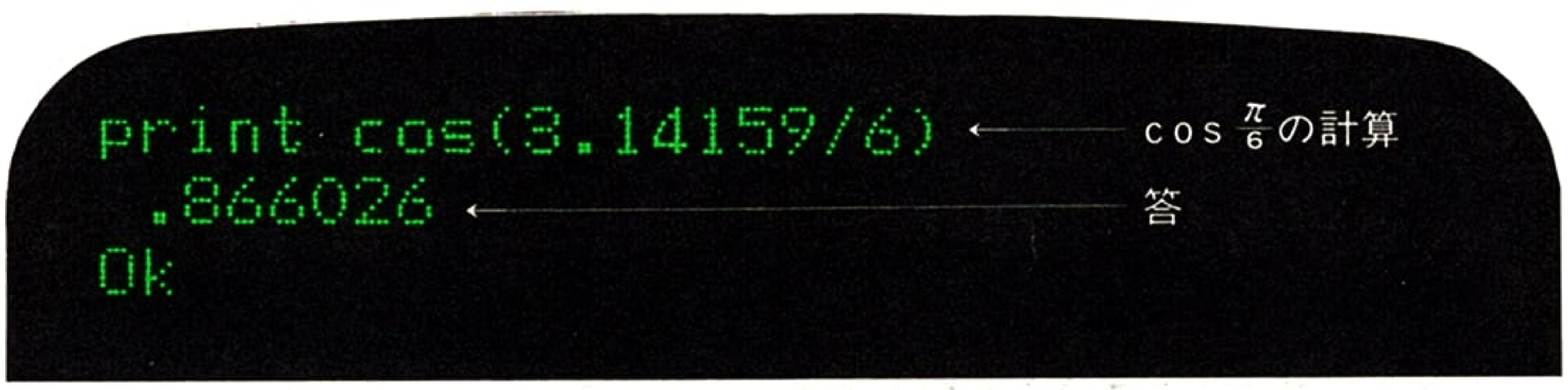
COS関数

次に、三角関数cos xの値を求めてみましょう。COS（コサイン）関数を使って、次のようにキーインします。

```
PRINT△COS(x) ⏎
```

← xの単位はラジアン

それでは、cos $\frac{\pi}{6}$ の値を求めてみましょう。次のようになります。



TAN関数

最後に、TAN（タンジェント）関数を使ってtan $\frac{\pi}{6}$ を求めてみましょう。

```
PRINT△TAN(x) ⏎
```

← xの単位はラジアン



第4章

N88-BASIC(N88DISK BASIC)入門

4

4・1 BASICとは

ＢＡＳＩＣとは何でしょうか？

ＰＣ-８８０１は、事務処理、技術計算、端末、教育、計測制御など、幅広く利用できます。しかし、日本語や英語でＰＣ-８８０１に話しかけても、ＰＣ-８８０１にはわかりません。ＰＣ-８８０１を動かすには、ＢＡＳＩＣ（ベーシック）と呼ばれる言語が必要です。

ＢＡＳＩＣ

それでは、ＢＡＳＩＣとは何でしょうか。

ＢＡＳＩＣとは、Beginner's All－purpose Symbolic Instruction Code（初心者向きで、あらゆる目的にあう記号を使った命令語）の略です。何かむずかしそうですね。しかし、心配はいりません。

あなたは、ＢＡＳＩＣ言語を使って、ＰＣ-８８０１のキーボードのキーをたたいて、ＰＣ-８８０１に話しかければよいのです。そうすると、ＰＣ-８８０１は、ディスプレイの画面を通してあなたに返答してきます。つまり、ＢＡＳＩＣという言語は会話型言語なのです。

どうです？　ＢＡＳＩＣ言語を使って、ＰＣ-８８０１と話をしてみたいとは思いませんか？　ところが、あなたは、もうＰＣ-８８０１と話しはじめているのです。「ウッソー」なんて言わないで、いままでのことを思い出してください。

あなたは、すでに、電源をＯＮにし、キー操作の練習をしました。また、デモプログラムをロードし実行してみました。ＰＣ-８８０１を電卓がわりに使ってみましたね。こうした中で、あなたは、しらずしらずのうちに、ＢＡＳＩＣ言語を使っていたのです。

あなたは、ＢＡＳＩＣ言語を使って、キーボードのキーをたたいてＰＣ-８８０１に話しかけていたわけです。それだからこそ、ＰＣ-８８０１は、ディスプレイの画面にグラフを書いたり、計算結果を表示したのです。もっとも、ときには、エラーメッセージも出てきましたね。

N_{88}－ＢＡＳＩＣとＮ－ＢＡＳＩＣ

ＢＡＳＩＣは、初心者用のどんな目的にも使える会話型言語です。しかし、パーソナルコンピュータのメーカーや機種によって、使えるＢＡＳＩＣがちがいます。

N_{60}－ＢＡＳＩＣ
Ｎ－ＢＡＳＩＣ
N_{88}－ＢＡＳＩＣ

ＮＥＣのパーソナルコンピュータの場合、ＰＣ-６００１にはN_{60}-ＢＡＳＩＣ、ＰＣ-８００１にはＮ－ＢＡＳＩＣ、そしてＰＣ-８８０１にはN_{88}－ＢＡＳＩＣとＮ－ＢＡＳＩＣを使います。

N－BASIC
N_{88}－BASIC

N－BASICは、PC-8001が持っているBASICです。N_{88}－BASICは、このN－BASICに多くの拡張機能を付け加えたBASICで、PC-8801のメインBASICです。

PC-8801はN_{88}－BASICとN－BASICの2つのBASICを持っています。ですから、PC-8801は、N_{88}－BASICで作られたプログラムでも、N－BASICで作られたプログラムでも使用できます。PC-8801では、PC-8001を使ってN－BASICで作られたプログラムも使用できるのです。便利ですね。

PC-8801においては、N_{88}－BASICとN－BASICは、必要に応じて切りかえて使用します。

N_{88}DISK－BASICとDISK－BASIC

N_{88} DISK－BASIC
DISK－BASIC

PC-8801では、ディスク（フロッピィディスクあるいはミニフロッピィディスク）を使う場合、ディスク用のBASIC（ディスクモードのBASIC）を使用します。ディスクモードのBASICには、N_{88}DISK－BASICとDISK－BASICの2つがあります。

N_{88}DISK－BASICは、N_{88}－BASICにディスクに関するBASICを加えたものです。なお、ディスクに関するBASICは、N_{88}DISK－BASIC用システムディスクに入っています。

同じように、DISK－BASICは、N－BASICにディスクに関するBASICを加えたものです。なお、ディスクに関するBASICは、DISK－BASIC用システムディスクに入っています。

2つのBASIC		2つのディスクモードのBASIC
●N_{88}－BASIC	——→	●N_{88}DISK－BASIC
●N－BASIC	——→	●DISK－BASIC

この章では、N_{88}－BASICとN_{88}DISK－BASICを取り上げます。N－BASICとDISK－BASICについては、第9章を参照してください。N_{88}－BASIC(N_{88}DISK－BASIC)とN－BASIC(DISK－BASIC)の切りかえについても、第9章を参照してください。

4・2 N₈₈ーBASICのスタート方法

N₈₈ーBASICのスタート順序

N₈₈ーＢＡＳＩＣ

N₈₈ーＢＡＳＩＣをスタートさせるには、すでに第０章の電源ＯＮで述べたように、次の順序で行ないます。

周辺装置の電源ＯＮ

(1) 周辺装置の電源をＯＮにします。

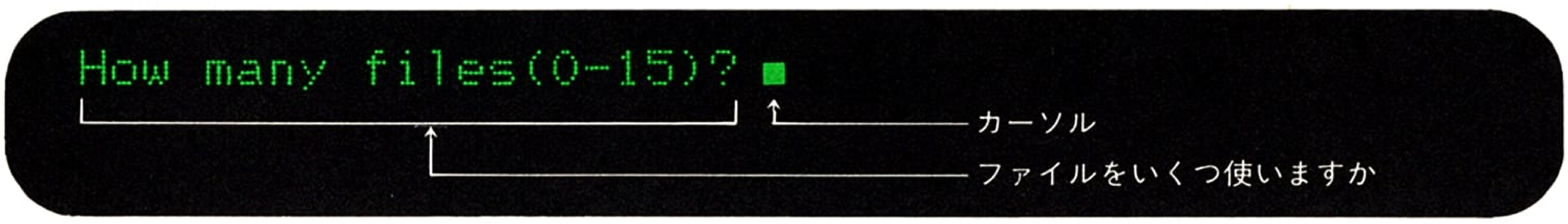
ＰＣ-８８０１の電源ＯＮ

(2) ＰＣ-８８０１の電源をＯＮにします。

この場合、ＰＣ-８８０１の電源スイッチの上にある電源表示用ＬＥＤが赤く点灯し、電源スイッチの下にあるＢＡＳＩＣ表示用ＬＥＤが緑色に点灯します。

(3) ディスプレイに次のメッセージが表示されます。

なお、このことは、途中でリセットボタンを押した場合も同じです。



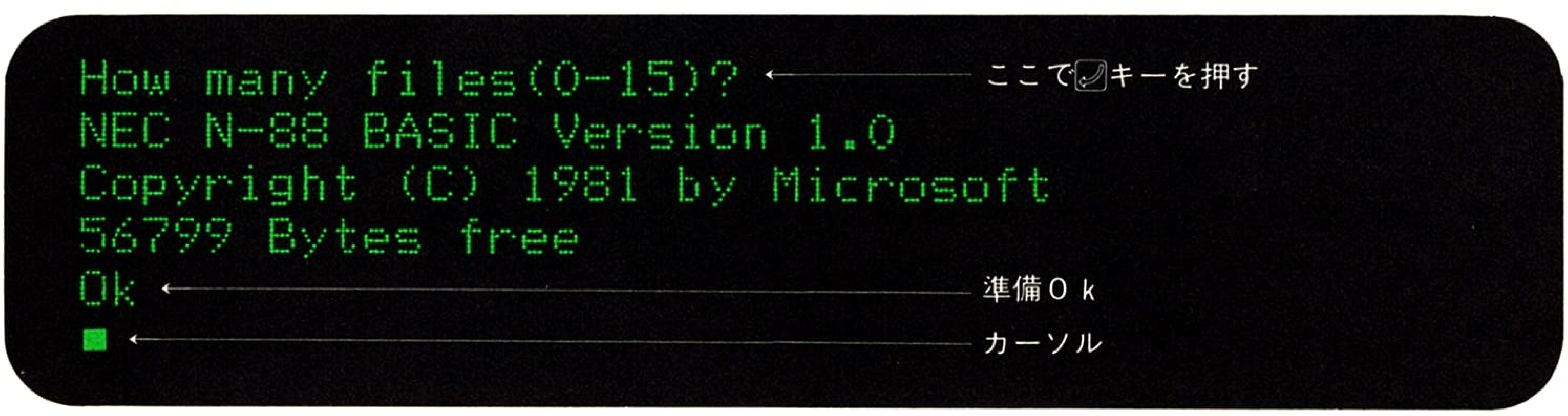
これは、「ファイルをいくつ使いますか」というＰＣ-８８０１からあなたへのメッセージです。

⏎キーを押す

(4) ⏎キーを押します。

N₈₈ーＢＡＳＩＣでは、ファイルを同時に最大15まで使用できます。数値をキーインせず、⏎キーを押すと、２が入力されたとみなします。

(5) 次のメッセージが表示されます。



これで、N₈₈ーＢＡＳＩＣがスタートできたことになります。

N₈₈ーＢＡＳＩＣがうまくスタートできないときは？

N₈₈ーＢＡＳＩＣがうまくスタートできない場合、まず全装置の電源をＯＦＦにし、マニュアルや取扱説明書をみながら、全装置の接続のしかたをチェックしてください。また、ＰＣ-８８０１の本体の背面にあるディップスイッチやジャンパスイッチを調べてください。

4・3　N88DISK－BASICのスタート方法

N88DISK－BASICのスタート順序

N88DISK－BASIC

N88ＤＩＳＫ－ＢＡＳＩＣをスタートさせるには、すでに第０章の電源ＯＮで述べたように、次の順序で行ないます。

周辺装置の電源ＯＮ

(1)　ディスクユニット（フロッピィディスクユニットやミニフロッピィディスクユニット）をはじめ、各周辺装置の電源をＯＮにします。

システムディスクのセット

(2)　システムディスクをディスクユニットのドライブ１にセットします。

電源ＯＮ
PC-8801の
リセットボタンを押す

(3)　ＰＣ－８８０１の電源をＯＮにして、リセットボタンを押します。すると、ディスクユニットのモータが回転し始めます。

(4)　システムディスクをセットしたドライブ１のアクセス表示用ＬＥＤが赤く点灯し、システムプログラムのロード（読み込み）が始まったことを示します。

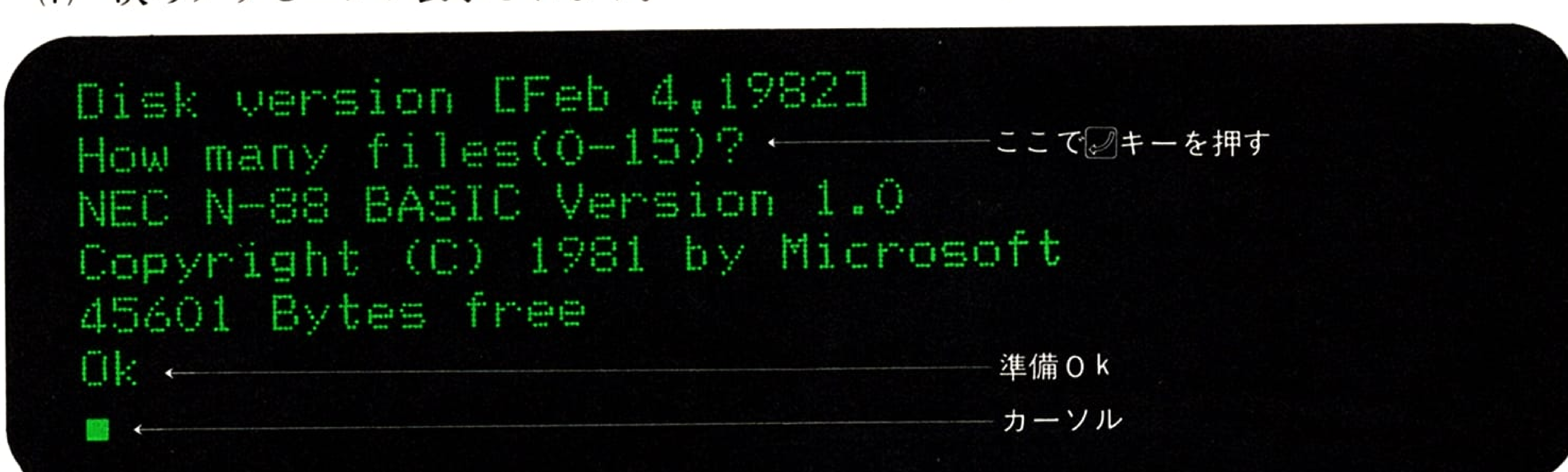
(5)　ドライブ１のアクセス表示用のＬＥＤが消灯し、ディスプレイに次のメッセージが表示されます。



キーを押す

(6)　キーを押します。

(7)　次のメッセージが表示されます。



これで、N88ＤＩＳＫ－ＢＡＳＩＣがスタートできたことになります。

N88DISK－BASICがうまくスタートできない場合

全装置の接続のしかた、ディップスイッチやジャンパスイッチをチェックしてください。

4・4 システムディスクのバックアップ

ディスクのクラッシュにそなえる

ディスクのクラッシュ

ディスク(フロッピィディスクやミニフロッピィディスク)に記録されているプログラムファイルやデータファイルは、こわれる(クラッシュする)ことがあります。途中で電源が切れたとか、まちがって必要なファイルを消してしまったとか、ディスクに磁石などの磁気の強いものを近づけてしまったとか、ディスクにはクラッシュがつきものです。

ディスクのバックアップ

そこで、重要な情報が記録されているディスク、とくにシステムディスクはコピーをとっておき、オリジナルのディスクはしまっておいて、コピーしたディスクを使うようにします。このように、ディスクのコピーをとっておいて、いつクラッシュしてもいいように準備しておくことを、ディスクのバックアップといいます。

バックアッププログラム

ｂａｃｋｕｐ.ｎ88
バックアッププログラム

システムディスクには、ｂａｃｋｕｐ.ｎ88というバックアッププログラムが入っています。

このバックアッププログラムｂａｃｋｕｐ.ｎ88をシステムディスクからＰＣ-８８０１にロードして実行すれば、ディスクのバックアップをすることができます。

それでは、システムディスクのバックアップをしてみましょう。

システムディスクの種類

N_{88}ＤＩＳＫ－ＢＡＳＩＣ用のシステムディスク

N_{88}ＤＩＳＫ－ＢＡＳＩＣ用システムディスクには、次のようなものがあります。

ディスクユニット		N_{88}ＤＩＳＫ－ＢＡＳＩＣ用システムディスク
● ８インチ標準フロッピィディスクユニット(ＰＣ-８８８１)	→	● ８インチ標準フロッピィシステムディスク(ＰＣ-８８８４)
● デュアルミニディスクユニット(ＰＣ-８０３１) ● シングルドライブ片面ミニディスクユニット(ＰＣ-８０３１－１Ｖ)	→	● 片面倍密度ミニフロッピィシステムディスク(ＰＣ-８８３４－１Ｗ)
● 両面倍密度デュアルミニディスクユニット(ＰＣ-８０３１－２Ｗ)	→	● 両面倍密度ミニフロッピィシステムディスク(ＰＣ-８８３４－２Ｗ)

8インチ標準フロッピィシステムディスクのバックアップ

8インチ標準フロッピィシステムディスクのバックアップは、次の順序で行ないます。

(1) 8インチ標準フロッピィディスクユニットをはじめ、各周辺装置の電源をONにします。

(2) システムディスクを8インチ標準フロッピィディスクユニットのドライブ1にセットし、新しいディスク（システムディスクにしたいディスク）をドライブ2にセットします。

システムディスク

(3) PC-8801の電源をONにし、リセットボタンを押します。

(4) ディスプレイに次のメッセージが表示されます。

```
Disk version [Feb 4,1982]
How many files(0-15)? ■
```

← ⏎キーを押す

(5) ⏎キーを押します。

(6) ディスプレイに次のメッセージが表示されます。

```
Disk version [Feb 4,1982]
How many files(0-15)?
NEC N-88 BASIC Version 1.0
Copyright (C) 1981 by Microsoft
45601 Bytes free
Ok
■
```

Ok ← 準備Ｏｋ

これで、N₈₈ＤＩＳＫ－ＢＡＳＩＣがスタートできました。

(7) ＦＩＬＥＳ（ファイルズ）コマンドを使って、システムディスクに入っているファイル名を画面に表示してみます。次のようにキーインします。なお、ＦＩＬＥＳは、小文字のｆｉｌｅｓでもかまいません。

ＦＩＬＥＳコマンド

```
files ⏎
```

← システムディスクに入っているファイル名を画面に表示

(8) 画面に次のように表示されます。ファイル名の中に、バックアップブログラムｂａｃｋｕｐ.ｎ88の名前がありますね。

バックアッププログラム
ｂａｃｋｕｐ.ｎ88

```
files
backup.n88 1    format.n88 1    setinf.n88 1    xfiles.n88 1    Quartz.    1
Cpaint.    1    DEMO   .   1    DEMOUT.    2
Ok
■
```

backup.n88 ← バックアッププログラム

LOADコマンド
backup.n88

(9)　LOAD（ロード）コマンドを使って、バックアッププログラムbackup.n88をシステムディスクからPC-8801にロードします。なお、 f・1 キーにload△"が定義されていましたね。次のようにキーインしましょう。

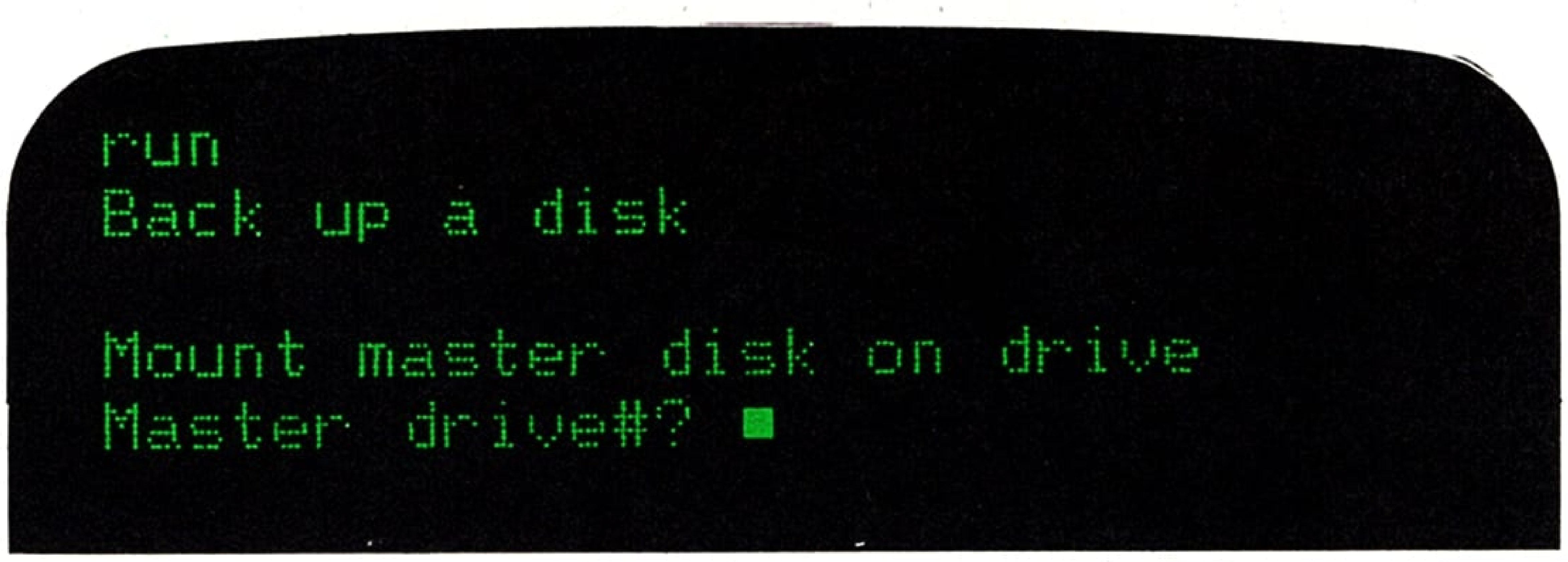
```
f・1  backup.n88"  ⏎
```

RUNコマンド

(10)　RUN（ラン）コマンドを使って、バックアッププログラムbackup.n88を実行します。なお、 f・5 キーにrun $^{C}_{R}$ が定義されていましたね。 f・5 キーを押しましょう。

```
f・5
```

(11)　すると、画面に次のメッセージが表示されます。



(12)　すでに、システムディスクをドライブ1にセットしてあるので、次のようにキーインします。

```
1 ⏎
```
システムディスクはドライブ１にセットずみ

(13)　画面に次のメッセージが表示されます。



(14)　すでに、新しいディスク(システムディスクにしたいディスク)をドライブ2にセットしてあるので、次のようにキーインします。

```
2 ⏎
```
新しいディスクはドライブ２にセットずみ

(15)　画面に次のメッセージが表示されます。



(16)　ドライブ番号を確認してきたので、ドライブ番号が正しければ次のようにキーインします。

```
y ⏎
```

もし、ドライブ番号がまちがっていれば、次のようにキーインしてもう一度(11)からやり直します。

```
n ⏎
```

(17)　ドライブ1とドライブ2のアクセス表示用ＬＥＤが順に点灯すると同時に、コピーしているトラック番号が画面に表示されます。

```
Copying track トラック番号
```
← コピーしているトラック番号の表示

(18)　システムディスクのバックアップが終了すれば、画面に次のように表示されます。

```
Completed.
Ok
■
```
← システムディスクのバックアップ終了

(19)　念のため、ＦＩＬＥＳコマンドを使って、オリジナルのシステムディスクとコピーしたシステムディスクに入っているファイル名を、それぞれ画面に表示してみましょう。この場合、コピーしたシステムディスクについては、それがドライブ2にセットしてあるので、ｆｉｌｅｓ△2⏎とキーインします。

ＦＩＬＥＳコマンド

```
files ⏎
```
← オリジナルのシステムディスクに入っているファイル名を画面に表示

```
files △ 2 ⏎
```
← コピーしたシステムディスクに入っているファイル名を画面に表示

△ ← スペースを１つあける

(20)　画面に次のように表示されます。

```
files
backup.n88 1    format.n88 1    setinf.n88 1    xfiles.n88 1    Quartz.    1
Cpaint.    1    DEMO   .   1    DEMOUT.    2
Ok
files 2
backup.n88 1    format.n88 1    setinf.n88 1    xfiles.n88 1    Quartz.    1
Cpaint.    1    DEMO   .   1    DEMOUT.    2
Ok
```
files ← オリジナルのシステムディスクに入っているファイル名を画面に表示

files 2 ← コピーしたシステムディスクに入っているファイル名を画面に表示

(21)　オリジナルのシステムディスクに入っているファイル名をプリンタに出力したければ、ＬＦＩＬＥＳ（エルファイルズ）コマン

ＬＦＩＬＥＳコマンド

ドを使えばよかったですね。なお、ＬＦＩＬＥＳは、小文字のｌｆｉｌｅｓでもかまいません。次のようにキーインします。

```
lfiles ⏎   ←―― オリジナルのシステムディスクに入っているファイル名をプリンタに出力
```

ドライブ２にセットしてあるコピーしたシステムディスクに入っているファイル名をプリンタに出力したければ、次のようにします。

```
lfiles △ 2 ⏎   ←―― コピーしたシステムディスクに入っているファイル名をプリンタに出力
       ↑―――――― スペースを１つあける
```

(22) プリンタには、次のように出力されます。オリジナルのシステムディスクでもコピーしたシステムディスクでも同じです。

```
backup.n88 1    format.n88 1    setinf.n88 1    xfiles.n88 1    Quartz.    1
Cpaint.    1    DEMO  .    1    DEMOUT.    2
```

以上で、新しいディスク（システムディスクにしたいディスク）がシステムディスクになりました。以後、オリジナルのシステムディスクはしまっておいて、コピーしたシステムディスクを使うようにしましょう。

ミニフロッピィシステムディスクのバックアップ

ミニフロッピィシステムディスクのバックアップは、８インチ標準フロッピィシステムディスクのそれとほぼ同じ順序です。８インチ標準フロッピィディスクユニットをミニフロッピィディスクユニットに読みかえれば、キーインとか画面に表示されるメッセージやプリンタへの出力はほぼ同じです。ちがうところは、(15)です。(15)を次のようにしてください。

(15) (14)で２⏎キーインすると、画面に次のメッセージが表示されます。

```
Do you need physical formatting(y/n)? ■
```

ここで、次のようにキーインします。

```
y ⏎
```

すると、画面に次のメッセージが表示されます。

```
Physical format a disk 2
Sure(y/n)? ■
```

これで、(16)へ続きます。

4・5　パソコンの2とおりの使い方

N₈₈－ＢＡＳＩＣあるいはN₈₈ＤＩＳＫ－ＢＡＳＩＣがスタートすると、ディスプレイの画面にはＯｋという文字が表示され、ＯｋのＯの文字の下でカーソルがブリンク（点滅）しますね。この状態になれば、ＰＣ-８８０１が使えるわけです。

ところで、一般に、パソコンの使い方には、ダイレクトモードとプログラムモードの２とおりの使い方があります。

ダイレクトモード

ある決められた内容をＰＣ-８８０１に命令するという意味をもつ言葉をコマンドといいましたね。ＢＡＳＩＣのコマンドをＢＡＳＩＣの文法にしたがって、キーボードから入力すると、ＰＣ-８８０１はそのとおり実行します。

このように、すぐ実行させてみるというパソコンの使い方が、ダイレクトモード（直接実行形）です。

ダイレクトモード

第３章「電卓がわりに使うには」では、ＰＲＩＮＴコマンドを使って、四則計算・複利計算・平方根の計算・関数計算の計算結果を画面に表示させましたね。これは、実は、ダイレクトモードでＰＣ-８８０１を使っていたわけです。

プログラムモード

パソコンを動かすための命令書のようなものをプログラムといいましたね。このプログラムを先に作って、あとで順序よく実行させるというパソコンの使い方が、プログラムモード（間接実行形）です。

プログラムモード

第２章「デモンストレーション」では、デモプログラムをロードして実行しましたね。この場合、デモプログラムはすでに作成ずみでしたが、これはプログラムモードでＰＣ-８８０１を使っていたわけです。

プログラムモードがパソコン本来の使い方です。

N₈₈－ＢＡＳＩＣあるいはN₈₈ＤＩＳＫ－ＢＡＳＩＣをスタートさせて、まず、簡単なプログラムを作成して、実行してみましょう。

習うより慣れろです。キーの押しまちがいやエラーメッセージを気にせず、先へ進みましょう。

4·6 プログラムの作成

元利合計プログラム

いよいよプログラムを作成しますが、まず、元金Pを年利率R%の複利でN年間預けた場合の元利合計Sを計算するプログラムを作成してみましょう。

元利合計プログラム

この元利合計プログラムを作成する手順を整理すれば、次のようになります。

(1)　キーボードから、元金・年利率（%）・年数をキーインする
(2)　元利合計を計算する
(3)　計算結果をディスプレイに表示する
(4)　プログラムの実行を終了する

この手順にしたがって元利合計プログラムを作成すれば、次のとおりです。

```
10 INPUT "ｶﾞﾝｷﾝ =";P
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
60 END
```

← 元利合計プログラム

行番号

この元利合計プログラムの各行について、上から順に見ていきましょう。なお、BASICのプログラムの各行は、常に行番号で始まります。

ＩＮＰＵＴ文

❶　10～30行目のＩＮＰＵＴ（インプット）文——ＩＮＰＵＴ文はデータをキーボードからキーインするためのもので、次の形をしています。

INPUT △ "メッセージ" ; 変数名 ← ＩＮＰＵＴ文

（; ← セミコロン　" " ← ダブルクォーテーション）

なお、変数とはいろいろな値をとることができる数をいい、その名前を変数名といいます。

代入文

❷40行目の代入文——代入文は、式の値を変数に代入するためのもので、次の形をしています。

変数名＝式 ← 代入文

40行目は、元利合計を計算し、その値を元利合計を意味する変数Sの値とするという意味です。

❸ 50行目のPRINT文――PRINT文は、ディスプレイに情報を出力するためのものです。ここでは、次の形をしています。なお、print△がf・8に定義されていましたね。　**PRINT文**

PRINT △ " メッセージ " ; 変数名

- PRINT文
- セミコロン
- ダブルクォーテーション

❹ 60行目のEND（エンド）文――END文は、プログラムの実行を終了させるためのものです。　**END文**

新しくプログラムを作成する場合

次に元利合計プログラムをキーインするのですが、新しくプログラムを作成する場合は、まず、NEW（ニュー）コマンドを使って、次のようにキーインします。なお、NEWは小文字のnewでもかまいません。　**NEWコマンド**

NEW ⏎ ← 新しくプログラムを作成！

NEWコマンドによって、現在PC-8801のメモリにあるプログラムはすべて消されます。

行番号を自動的に発生させるには

プログラムの各行は行番号で始まるので、プログラムの各行をキーインするとき、まず行番号をキーインしなければなりません。しかし、AUTO（オート）コマンドを使えば、行番号を自動的に発生させることができます。　**AUTOコマンド**

行番号を10、20、30、……のように10から始まって10きざみで発生させるには、次のようにします(小文字のautoでもよい)。

AUTO ⏎ ← 行番号を10から始まって10きざみで自動的に発生させる

しかし、 f･2 キーにauto△が定義されていますので、次のようにキーインしてみましょう。

f･2 ⏎

画面には、次のように表示されます。

```
new
Ok
auto
10
```

プログラムのキーイン

それでは、元利合計プログラムをキーインしてください。

なお、アルファベットは小文字でキーインしてもかまいません。50行目のｐｒｉｎｔ△については、ｐｒｉｎｔ△がｆ・８に定義されていましたので、 SHIFT キーを押しながら f·3 キーを押せばよかったですね。

また、カナ文字のキーがなかなか見つからないでしょうが、がんばって探してください。

各行について、キーインが終わったら、↵キーを押します。↵キーを押さなければ、画面に表示されていてもその内容はＰＣ-８８０１のメモリに記憶されません。各行について、キーインが終わったら、必ず、↵キーを押しましょう。

↵キーを押すと、次の行番号が自動的に発生しますね。すばらしいですね。

↵キーを押す前であれば、キーの押しまちがいなどエラーを修正することができます。エラーの修正には、カーソル移動キー（↑、←、→、↓）や DEL キー、 INS キーを使えばよかったですね。

↵キーを押してしまってからエラーに気がついた場合には、そのエラーはあとで修正することにしてそのままにしておき、とにかくキーインを続けましょう。

60行目のＥＮＤ文のキーインが終わって↵キーを押すと、行番号70が自動的に発生します。

STOP キー

ここで、 STOP （ストップ）キーを押して行番号の自動発生をストップさせます。

さあ、次のようにキーインできましたか？

```
auto
10 input "ｶﾞﾝｷﾝ =";p
20 input "ﾈﾝﾘ(%) =";r
30 input "ﾈﾝｽｳ =";n
40 s=p*(1+r/100)^n
50 print "ｶﾞﾝﾘｺﾞｳｹｲ =";s
60 end
70
```

プログラムリストの出力

CLR キー

ＬＩＳＴコマンド

ここで、 CLR （クリア）キーを押して、画面をクリアしましょう。そして、ＬＩＳＴ（リスト）コマンドを使って、プログラムの全部（プ

ログラムリスト）をディスプレイに表示してみましょう。次のようにします。なお、LISTは小文字のlistでもかまいません。

```
LIST ↵  ←―――― プログラムリストをディスプレイに出力
```

しかし、list△が [f·4] キーに定義されていますので、次のようにキーインしましょう。

```
[f·4] ↵
```

画面には、次のように表示されます。小文字だったアルファベットが大文字に変っていますね。また、最後に発生された行番号70は消えていますね。

```
list
10 INPUT "ｶﾞﾝｷﾝ =";P
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
60 END
Ok
```

（10～60行）← 小文字だったアルファベットが大文字に変っている

（Ok）← 行番号70は消えている

LLISTコマンド

プログラムリストをプリンタに出力するには、LLIST（エルリスト）コマンドを使って、次のようにします。LLISTは小文字のllistでもかまいません。

```
LLIST ↵  ←―――― プログラムリストをプリンタに出力
```

次のようにキーインしましょう。

```
l [f·4] ↵
```

プリンタの出力は、次のとおりです。

```
10 INPUT "ｶﾞﾝｷﾝ =";P
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
60 END
```

エラー

プログラムを作成したら、必ずプログラムリストを出力して、プログラムにエラーがないかどうかをチェックしましょう。

エラーがあれば、カーソル移動キー（[↑]、[←]、[→]、[↓]）や [DEL] キー、[INS] キーを使って修正します。修正したら、必ず↵キーを

押します。⏎キーを押さなければ修正したことにはなりません。

なお、このようにプログラムのエラーを探し出して修正する作業を

デバッグ

デバッグ（虫取り）といいます。

いま、プログラムリストをディスプレイに出力したら、次のようであったとしましょう。

```
list
10 INPUT "ｶﾞﾝｷﾝ =";P
20 IMPUT "ﾈﾝﾘ(%)  =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
60 END
Ok
```

← IMPUTのMがエラー

20行目のIMPUTのMがエラーですね。MをNに修正しましょう。まず、↑キーと→キーを使って、カーソルを20行目のMの上に動かしましょう。

```
list
10 INPUT "ｶﾞﾝｷﾝ =";P
20 IMPUT "ﾈﾝﾘ(%)  =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
60 END
Ok
```

← ↑キーと→キーを使ってカーソルを20行目のMの上に動かし、Nキーを押し、⏎キーを押す

ここで、Nキーを押せばMがNに変わります。そして、⏎キーを押します。⏎キーを押さなければ、画面上の20行目のIMPUTのMがNに変わっただけで、PC-8801のメモリの中にある元利合計プログラムの20行目のMはNに変わりません。修正したら必ず⏎キーを押します。忘れないでください。

プログラムのエラーの修正が終わったら、CLRキーを押して画面をクリアし、LISTコマンドあるいはLLISTコマンドを使ってもう一度プログラムリストを出力して、ちゃんと修正されているかどうか確認しましょう。

```
list
10 INPUT "ｶﾞﾝｷﾝ =";P
20 INPUT "ﾈﾝﾘ(%)  =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
60 END
Ok
```

← IMPUTのMがNに修正されている

4·7 プログラムの実行

元利合計プログラムを実行しよう

それでは、元利合計プログラムを実行してみましょう。

プログラムの実行には、RUN（ラン）コマンドを使って、次のようにすればよかったですね。なお、RUNは小文字のrunでもかまいません。

RUNコマンド

RUN ↵	← プログラムの実行

しかし、run $^{C}_{R}$ が f·5 キーに定義されていたので、f·5 キーを押せばよかったですね。f·5 キーを押してみましょう。

f·5	← f·5 キーを押す。↵キーを押す必要はありません。

画面には、次のように、元金の入力をうながすメッセージと？（クエッションマーク）が表示されます。

```
run
ｶﾞﾝｷﾝ =? ■
```
← 元金はいくらですか？

ここで、元金として100(万円) をキーボードからキーインしてみましょう。

100 ↵	← 元金は100万円です

↵キーを押すと、元金を意味する変数Pに100が代入されます。次に画面には、年利率の入力をうながすメッセージと？が表示されます。

```
run
ｶﾞﾝｷﾝ =? 100
ﾈﾝﾘ(%) =? ■
```
← 年利率は何%ですか？

年利率として、6（%）をキーインしてみましょう。

6 ↵	← 年利率は6%です

年利率を意味する変数Rに6が代入されます。

次の画面には、年数の入力をうながすメッセージと？が表示されます。

```
run
ｶﾞﾝｷﾝ =? 100
ﾎﾝﾘ(%) =? 6
ﾎﾝｽｳ =?
```

← 何年預けますか？

年数として10（年）をキーインしてみましょう。

```
10 ⏎
```

← 10年預けます。

年数を意味する変数Nに10が代入されます。

次に、元利合計が計算され、その値が元利合計を意味する変数Ｓに代入されます。

そして、ディスプレイにメッセージと元利合計Ｓの値が表示され、実行が終わり、Ｏｋが表示されます。

```
run
ｶﾞﾝｷﾝ =? 100
ﾎﾝﾘ(%) =? 6
ﾎﾝｽｳ =? 10
ｶﾞﾝﾘｺﾞｳｹｲ = 179.085
Ok
```

「179.085」← 元利合計

元金 100万円を年利率６％で10年間預ければ、元利合計は 179.085万円になります。

以上のようになりましたか？　エラーメッセージは出ませんでしたか？

いま、プログラムリストをディスプレイに出力したら、次のようであったとしましょう

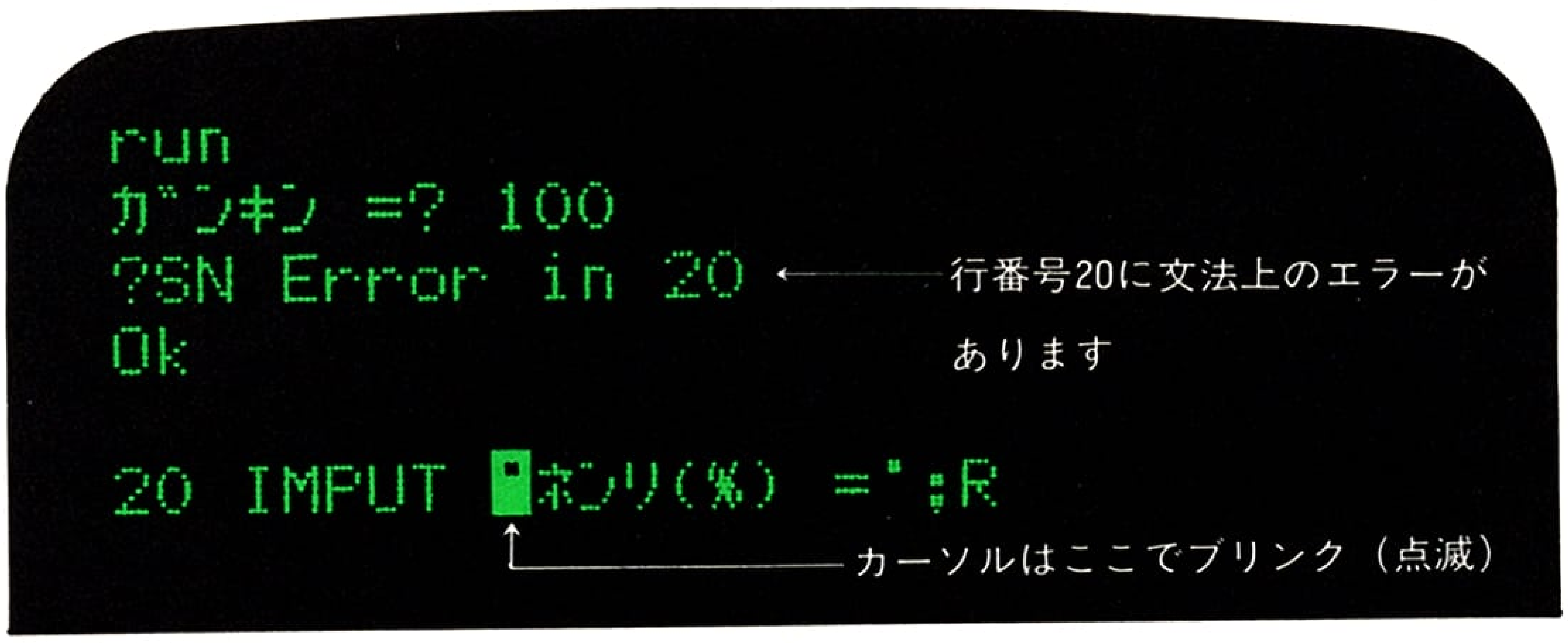
```
list
10 INPUT "ｶﾞﾝｷﾝ =";P
20 IMPUT "ﾎﾝﾘ(%) =";R
30 INPUT "ﾎﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
60 END
Ok
```

20行 ← ＩＭＰＵＴのＭがエラー

このプログラムをこのまま実行するとどうなるでしょうか。

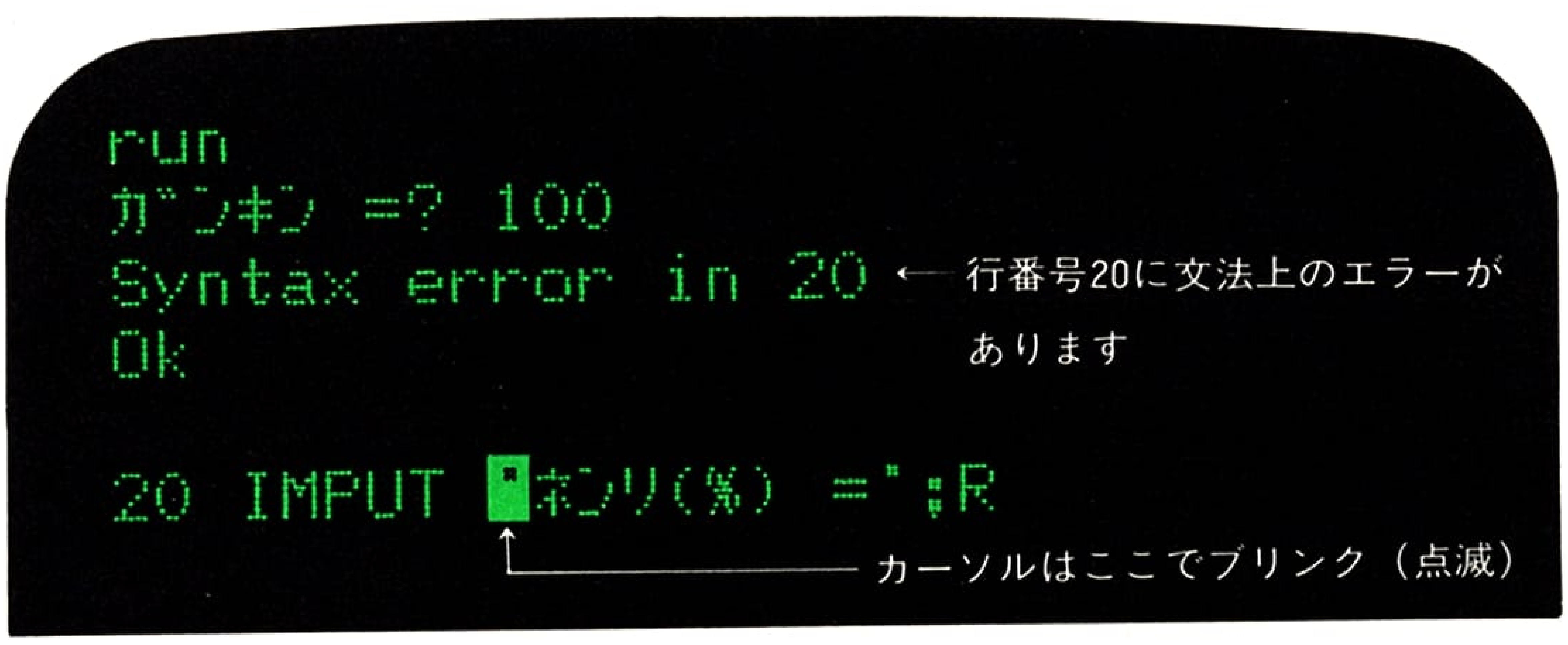
N_{88}−ＢＡＳＩＣ エラーメッセージ

N_{88}−ＢＡＳＩＣでは、次のようなエラーメッセージが表示されます。



N88DISK-BASIC
エラーメッセージ

N88ＤＩＳＫ－ＢＡＳＩＣでは、次のようなエラーメッセージが表示されます。



?SN Error
Syntax error
エラーメッセージ

画面に「？ＳＮ　Ｅｒｒｏｒ　ｉｎ　20」あるいは「Ｓｙｎｔａｘ　ｅｒｒｏｒ　ｉｎ　20」が表示されましたね。これは、「行番号20に文法上のエラーがありますよ」というＰＣ-８８０１からあなたへのエラーメッセージです。

そして、画面にはＯｋの表示のあと、１行おいて、行番号20の文（ステートメント）が表示されています。

行番号20をみれば、ＩＭＰＵＴのＭがエラーであることがわかります。そこで、このＭをＮに修正します。次のようにすればよいですね。

まず、←キーを使ってカーソルをＩＭＰＵＴのＭの上に動かします。そして、Ｎキーを押し↵キーを押します。



エラーの修正が終わったら、CLRキーを押して画面をクリアし、ＬＩＳＴコマンドあるいはＬＬＩＳＴコマンドを使ってもう一度プログラムリストを出力し、ちゃんと修正されているかどうか確認します。確認がすんだら、f･5キーを押して元利合計プログラムを実行します。

4・8 プログラムのセーブ

元利合計プログラムをセーブしてみよう

ＰＣ-８８０１本体の電源をＯＦＦにしたり、リセットボタンを押すと、あなたがキーインした元利合計プログラムは消えてしまいます。そこで、ＰＣ-８８０１のメモリの中にある元利合計プログラムをカセットテープやディスクにセーブ（書き込む）しておきます。そうすれば、元利合計プログラムを使用するときに、カセットテープやディスクからＰＣ-８８０１のメモリに元利合計プログラムをロード(読み込む）して、ラン（実行）すればよいわけです。

セーブ
ロード
ラン（実行）

それでは、元利合計プログラムをカセットテープやディスクにセーブしてみましょう。

元利合計プログラムをカセットテープにセーブするには

カセットテープレコーダがＰＣ-８８０１に接続されていますね。

まず、録音してもよいカセットテープをカセットテープレコーダにセットします。

次に、録音（ＲＥＣ）ボタンと再生（ＰＬＡＹ）ボタンを同時に押します。

しかし、接続ケーブルの黒色プラグがカセットテープレコーダのリモート（ＲＥＭＯＴＥ）端子に接続してあれば、カセットテープは回り始めません。そこで、ＭＯＴＯＲ（モータ）文を使って、カセットテープレコーダのモータをＯＮにします。ＭＯＴＯＲ文を使えば、カセットテープレコーダのモータがＯＮの状態であればＯＦＦに、ＯＦＦの状態であればＯＮになります。次のようにキーインしましょう。

ＭＯＴＯＲ文

なお、ＭＯＴＯＲは小文字のｍｏｔｏｒでもかまいません。

```
MOTOR ⏎   ←――― カセットテープレコーダのモータＯＮ
```

すると、カセットテープは回り始めます。

カセットテープの始めの部分には、クリアフィーダという磁性体の塗ってない部分があります。この部分には記録できませんから、プログラムをセーブする前にスキップしておく必要があります。また、前のプログラムとの区別をつける意味からも、まず、ＭＯＴＯＲ文でカセットテープを少し回しておきましょう。

数秒したら、ＭＯＴＯＲ文でカセットテープレコーダのモータをＯＦＦにします。次のようにキーインします。

```
MOTOR ⏎ ←―――― カセットテープレコーダのモータOFF
```

すると、カセットテープが止まります。これで、セーブの準備ができたわけです。

PC-8801のメモリの中にあるプログラムをカセットテープにセーブするには、SAVE（セーブ）コマンドを使って、次のようにキーインします。なお、SAVEは小文字のsaveでもかまいませせん。

SAVEコマンド

```
SAVE △ "CAS:プログラムファイル名" ⏎ ←―― プログラムをカセットテープにセーブ
                  └―――――――――――――――――― プログラムファイルの名前
       └――――――――――――――――――――――――――― スペースを１つあける
```

ここで、CASはcassette　tape（カセットテープ）の最初の３文字だと考えましょう。なお、CASは小文字のcasでもかまいません。

SAVEコマンドによって、PC-8801のメモリの中にあるプログラムがカセットテープにセーブされます。



なお、save△"はf・6に定義されていますので、（SHIFT）キーを押しながら（f･1）キーを押せばよいですね。

それでは、元利合計プログラムをカセットテープにセーブしましょう。録音ボタン（REC）と再生（PLAY）ボタンは押されたままですね。プログラムファイル名をGanri　Goukeiの頭文字をとってGGとしましょう。次のようにキーインします。

```
(SHIFT) + (f･1) cas:GG" ⏎ ←―― 元利合計プログラムGGをカセットテープにセーブ
```

すると、カセットテープが回り始めて、PC-8801のメモリの中の元利合計プログラムがカセットテープにセーブされます。

セーブが終わると、画面にOkが表示されて、カセットテープは自動的に止まります。

セーブが終わったら、巻き戻し(REW)ボタンを押してカセットテープを巻き戻します。この場合、カセットテープが巻き戻せなければ、次のようにキーインしてから、巻き戻し(REW)ボタンを押します。

```
MOTOR ←―――― カセットテープレコーダのモータON
```

以上の手順を画面で見ると、次のようになります。

```
motor        ← カセットテープレコーダのモータON
Ok
motor        ← カセットテープレコーダのモータOFF
Ok
save "cas:GG"   ← 元利合計プログラムGGを
Ok                カセットテープにセーブ
```

接続ケーブルの黒色プラグをカセットテープレコーダのリモート（REMOTE）端子に接続しない場合とか、リモートの効かないテープレコーダの場合、テープレコーダのスタートやストップは自分で行う必要があります。このような場合は、まず、次のようにキーインします。ただし、⏎キーはまだ押してはいけません。

```
SHIFT + f·1 cas:GG"   ← ⏎キーはまだ押してはいけません
```

画面には、次のように表示されます。

```
save "cas:GG" ■   ← ⏎キーを押していないので、カーソルはまだここにある
```

次に、録音（REC）ボタンと再生（PLAY）ボタンを同時に押し、カセットテープを少し回します。そして、⏎キーを押し、元利合計プログラムGGをセーブします。

セーブが終わると、画面にOkが表示されます。ただし、カセットテープは回り続けます。

```
save "cas:GG"
Ok
```

画面にOkが表示されたら、ストップ（STOP）ボタンを押してカセットテープを止めます。そして、巻き戻し（REW）ボタンを押してカセットテープを巻き戻します。

カセットテープにプログラムをセーブした場合は、必ず、正しくセーブされたかどうかをチェックします。PC-8801のメモリの中のプログラムとカセットテープにセーブされたプログラムを比較して、正しくセーブされたかどうかを確認することをベリファイといいます。

ベリファイ
LOAD？

ベリファイするには、LOAD？（ロードクェッションマーク）コマンドを使って、次のようにキーインします。なお、LOADは小文字のloadでもかまいません。

```
LOAD? △ "CAS: プログラムファイル名   ← ベリファイ
      ↑
      └─ スペースを1つあける
```

それでは、元利合計プログラムのベリファイをしましょう。次のようにキーインします。まだ⏎キーを押してはいけません。

```
load? △"cas:GG"
```

↑ スペースを1つあける　← まだ⏎キーを押してはいけない

画面には、次のように表示されます。

```
load? "cas:GG" ■
```

↑ ⏎キーを押していないので、カーソルはまだここにある

次に、カセットテープレコーダの再生（ＰＬＡＹ）ボタンを押し、つづいて⏎キーを押します。すると、カセットテープが回りはじめます。

元利合計プログラムＧＧが見つかると、画面に次のように表示されます。

```
Found:GG
```

← 元利合計プログラムＧＧ発見

ベリファイがはじまると、画面の右上に＊（アスタリスク）がブリンク（点滅）し、ベリファイ中であることを示します。

ベリファイが終わると、ブリンクしていた＊が止まり、画面のＦｏｕｎｄの下にＯｋが表示されます。



Ｏｋが表示されれば、ＰＣ-８８０１のメモリの中のプログラムの内容とカセットテープにセーブされたプログラムの内容が一致したことになり、正しくセーブされたことになります。

もしどこか違ったところがあると、Ｂａｄが表示されます。



この場合は、カセットテープを巻き戻して、もう一度セーブからやり直します。

元利合計プログラムをディスクにセーブするには

N88DISK－BASICがスタートしていますね。

ＰＣ-８８０１のメモリの中にあるプログラムをディスク(フロッピィディスクあるいはミニフロッピィディスク）にセーブするには、ＳＡＶＥ（セーブ）コマンドを使って、次のようにキーインします。

ＳＡＶＥコマンド

なお、ＳＡＶＥは小文字のｓａｖｅでもかまいません。

SAVE △"プログラムファイル名"⏎ ← プログラムをディスクにセーブ

- プログラムファイル名 ← プログラムファイルの名前
- △ ← スペースを１つあける

ＳＡＶＥコマンドによって、ＰＣ-８８０１のメモリの中にあるプログラムがデイスクにセーブされます。



なお、ｓａｖｅ△〝がｆ・６に定義されているので、SHIFTキーを押しながらf･1キーを押せばよいですね。

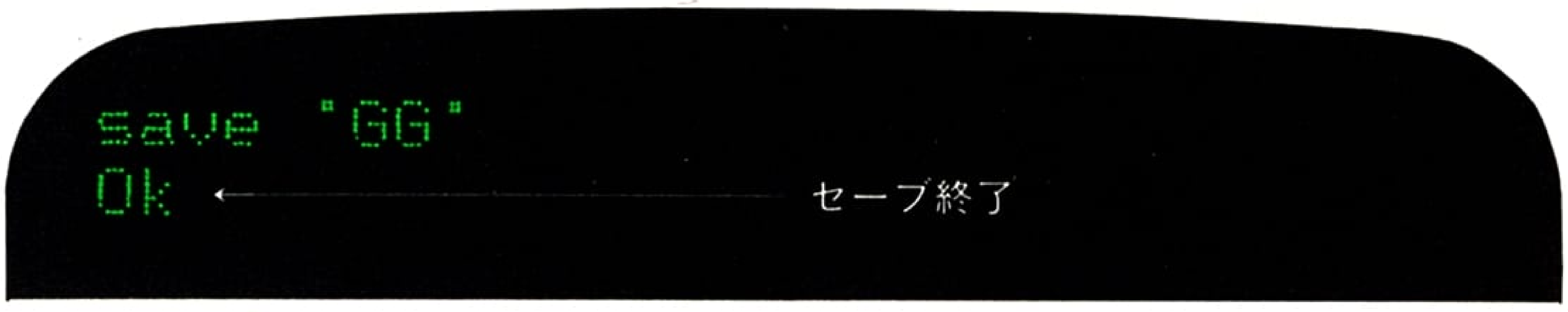
それでは、元利合計プログラムＧＧをドライブ１のディスクにセーブしましょう。プログラムファイル名をＧＧとします。次のようにキーインします。

SHIFT ＋ f･1 GG"⏎ ← 元利合計プログラムＧＧをドライブ１のディスクにセーブ

この場合、〝ＧＧ〟を〝１：ＧＧ〟としてもいいのですが、めんどうですね。

⏎キーを押すと、ドライブ１のアクセス表示用ＬＥＤが点灯し、プログラムのセーブが始まったことを示します。

セーブが終わると、ドライブ１のアクセス表示用ＬＥＤが消灯し、画面のｓａｖｅの下にＯｋが表示されます。



なお、ドライブ２のディスクにセーブするには、次のようにキーインします。

SHIFT ＋ f･1 2:GG"⏎ ← 元利合計プログラムＧＧをドライブ２のディスクにセーブ

ディスクを使えば、カセットテープを使うのと比較して、速く正確にプログラムをセーブすることができます。

ＦＩＬＥＳコマンド

元利合計プログラムがディスクにセーブされたかどうかを確認したければ、ＦＩＬＥＳコマンドを使ってファイル名を画面に表示してみればよいのです。

ドライブ１のディスクに入っているファイル名を画面に表示するには、次のようにキーインします。

```
files ⏎
```
←ドライブ１のファイル名を画面に表示

この場合、ｆｉｌｅｓ△１とキーインしてもいいのですが、めんどうですね。

⏎キーを押すと、次のようにファイル名が画面に表示されます。元利合計プログラム名ＧＧがありますね。



なお、ドライブ２のディスクに入っているファイル名を画面に表示するには、ｆｉｌｅｓ△２とキーインします。

ＬＦＩＬＥＳコマンド

ファイル名をプリンタに出力するには、ＬＦＩＬＥＳコマンドを使います。

ドライブ１のディスクに入っているファイル名をプリンタに出力するには、次のようにキーインします。ｌｆｉｌｅｓ△１でも同じことです。

```
lfiles ⏎
```
←ドライブ１のファイル名をプリンタに出力

なお、ドライブ２のディスクに入っているファイル名をプリンタに出力するには、ｌｆｉｌｅｓ△２とキーインします。

プリンタには、次のように出力されます。やはり、元利合計プログラム名ＧＧがありますね。

```
backup.n88 1    format.n88 1    setinf.n88 1    xfiles.n88 1    Quartz.    1
Cpaint.    1    DEMO  .    1    DEMOUT.    2    GG    .    1
```
元利合計プログラムＧＧ

4・9 プログラムのロード

元利合計プログラムをロードしてみよう

元利合計プログラムＧＧは、すでに、カセットテープあるいはディスクにセーブされています。そこで、元利合計プログラムを使用したいときには、カセットテープあるいはディスクからＰＣ-８８０１のメモリに元利合計プログラムをロードして、ラン（実行）すればよいわけです。

ＬＯＡＤコマンド

なお、プログラムをロードするには、ＬＯＡＤコマンドを使えばよかったですね。また、ｌｏａｄ△"が f・1 キーに定義されていましたね。

それでは、元利合計プログラムをカセットテープやディスクからロードしてみましょう。

元利合計プログラムをカセットテープからロードするには

次のようにキーインします。

```
f・1 cas:GG"
```
← 元利合計プログラムＧＧをロード

そして、カセットテープレコーダの再生（ＰＬＡＹ）ボタンを押し、つづいて⏎キーを押します。

画面には次のように表示されます。

```
load "cas:GG"
Found:GG
Ok
```
load "cas:GG" ← 元利合計プログラムＧＧをロード

アスタリスク →（右上の ＊）

元利合計プログラムをディスクからロードするには

次のようにキーインします。

```
files ⏎
```
← ドライブ１のファイル名を画面に表示

```
f・1 GG" ⏎
```
← 元利合計プログラムＧＧをロード

画面には、次のように表示されます。

```
files
backup.n88 1    format.n88 1    setinf.n88 1    ×files.n88 1    Quartz.    1
Cpaint.    1    DEMO   .   1    DEMOUT.    2    GG    .    1
Ok
load "GG"
Ok
```
files ← ドライブ１のファイル名を画面に表示

GG ← 元利合計プログラムＧＧ

load "GG" ← 元利合計プログラムＧＧをロード

4・10 プログラムの修正

元利合計プログラムのリスト

すでに、カセットテープあるいはディスクから元利合計プログラムGGがロードされていますね。

ここで、元利合計プログラムGGのリストを画面に表示してみましょう。プログラムリストを画面に表示するには、ＬＩＳＴコマンドを使いますが、次のようにキーインすればよかったですね。

ＬＩＳＴコマンド

```
(f･4) ⏎  ← プログラムリストをとる
```

画面には、次のように表示されます。

```
list  ← プログラムリストをとる
10 INPUT "ｶﾞﾝｷﾝ =";P
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
60 END
Ok
```

プログラムに注釈を入れるには

プログラムに注釈（コメント）を入れるには、ＲＥＭ（レム）文を使います。なお、ＲＥＭは小文字のｒｅｍでもかまいません。

ＲＥＭ文

```
REM △ コメント ⏎
     │   └──────── 注釈（コメント）
     └──────────── スペースを１つあける
```

行番号３として、注釈（コメント）の文をキーインすることにしましょう。

```
3 △ rem △ *** ｶﾞﾝﾘｺﾞｳｹｲ *** ⏎
  └─────┴──────── スペースを１つあける
```

プログラム上で画面をクリアするには

(CLR)（クリア）キーを押せば画面をクリアできますが、ＣＬＳ（クリアスクリーン）文を使えばプログラム上で画面をクリアできます。なお、ＣＬＳは小文字のｃｌｓでもかまいません。

ＣＬＳ文

```
CLS ⏎  ← 画面クリア
```

行番号５として、次のようにキーインしてみましょう。

```
5 cls⏎
```

画面クリア

ここで、プログラムリストを画面に表示してみましょう。次のようにキーインします。

```
(f·4) ⏎
```

プログラムリストを画面に表示

画面には、次のように表示されます。

```
list
3 REM *** ｶﾞﾝﾘｺﾞｳｹｲ ***
5 CLS
10 INPUT "ｶﾞﾝｷﾝ =";P
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
60 END
Ok
```

ＲＥＭ文のインサート

ＣＬＳ文のインサート

行番号３および５の文（ステートメント）が行番号10のＩＮＰＵＴ文の前にインサート（挿入）されましたね。

ＰＣ－８８０１は、行番号の小さい順に、１行１行おぼえていきます。したがって、ある行とその次の行の間に１行インサートしたければ、まだ使っていない行番号をつけてその行をキーインし、⏎キーを押せばよいわけです。

このように、プログラムを修正する場合、新しく行をインサートしたいことがたびたびおこってきます。そこで、プログラムを作成する場合に、あらかじめ行のインサートを考えて、行番号を10きざみにするなどして、間隔をとっておくのです。

それでは、このプログラムを実行してみましょう。

```
(f·5)
```

実行

画面には、次のように表示されます。

```
ｶﾞﾝｷﾝ =? 100
ﾈﾝﾘ(%) =? 6
ﾈﾝｽｳ =? 10
ｶﾞﾝﾘｺﾞｳｹｲ = 179.085
Ok
```

元利合計の繰り返し計算

ここで、 (f･5) キーを押せば、もう一度、元利合計の計算ができます。元金、年利、年数を変えて、元利合計を繰り返し計算してみましょう。

実際に元利合計の計算をする場合には、このように、元金、年利、年数のインプットデータを少しずつ変えて何回も続けて実行することが多いと思います。

そこで、そのつど (f･5) キーを押して実行するのではなく、プログラムを修正して、元利合計の繰り返し計算ができるようなプログラムにしてみましょう。

データをインプットして、元利合計を計算し、その結果をディスプレイに表示したら、元金データのインプットにもどればいいですね。

このように、指定した行へジャンプするには、ＧＯＴＯ（ゴートゥ）文を使います。

ＧＯＴＯ文

ＧＯＴＯ文は、次の形をしています。なお、ＧＯＴＯは、小文字のｇｏｔｏでもかまいません。

> GOTO △ 行番号 ⏎
>
> （△：スペースを１つあける）

また、ｇｏ△ｔｏ△が (f･3) キーに定義されています。

次のようにキーインしましょう。

> 55 △ (f･3) 10 ⏎ ← 行番号10へ飛べ
>
> （△：スペースを１つあける）

行番号55は、50のＰＲＩＮＴ文と60のＥＮＤ文の間の行番号です。51～59の範囲内であれば、何番でもかまいません。

プログラムリストをとってみましょう。

```
list
3 REM *** ｶﾞﾝﾘｺﾞｳｹｲ ***
5 CLS
10 INPUT "ｶﾞﾝｷﾝ =";P
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
55 GOTO 10
60 END
Ok
```

（55 GOTO 10 ← ＧＯＴＯ文のインサート）

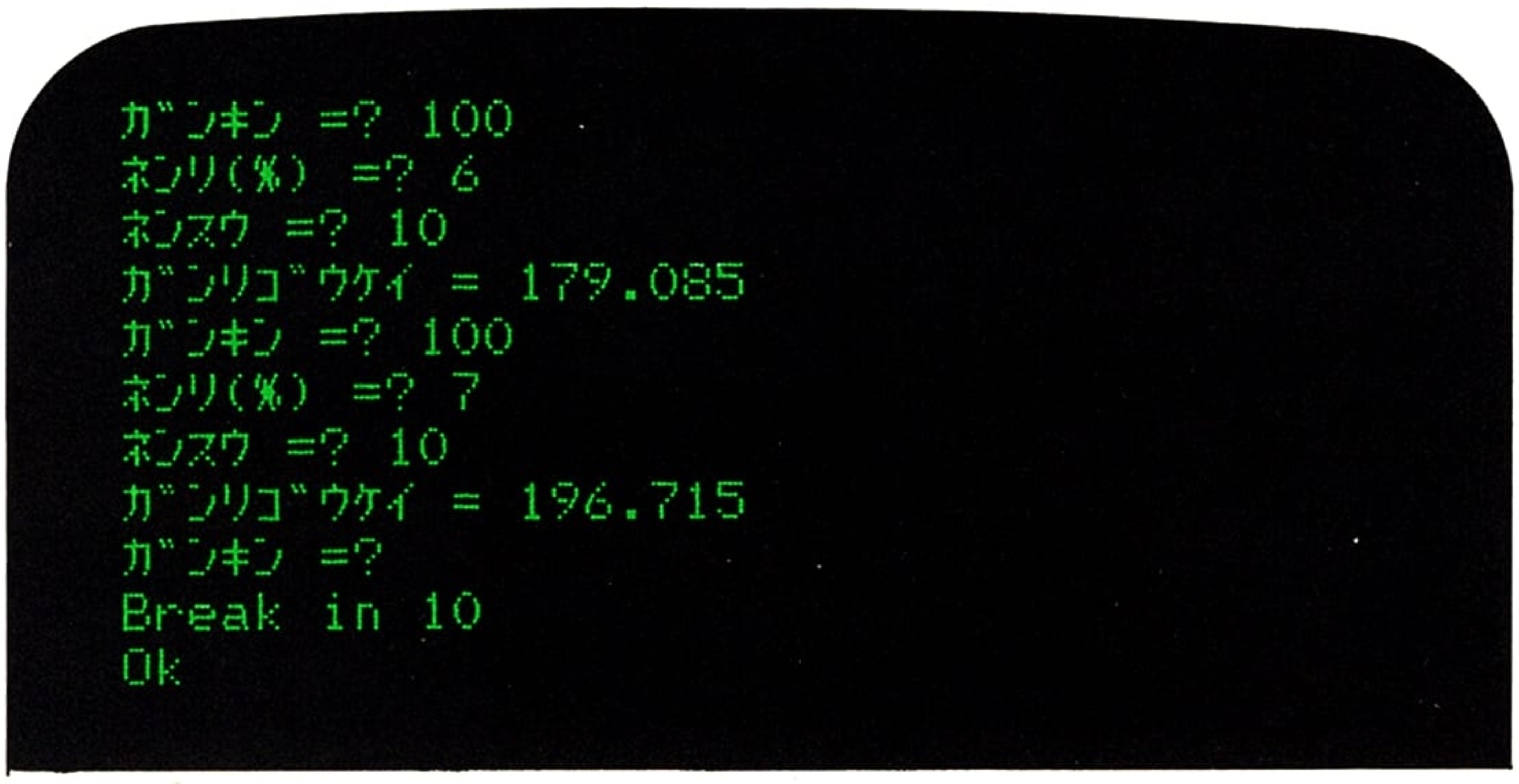
それでは、このプログラムを実行してみましょう。

繰り返し元利合計を計算できますね。しかし、何度でも元金データのインプットにもどってしまい、このプログラムは終わりませんね。どうしたらよいでしょうか。

プログラムの実行をストップさせるには

STOP キー

このプログラムの実行をストップさせるには、どうすればよいでしょうか。STOP キーを押せばよいのです。STOP キーを押してみましょう。次のようになります。



「Ｂｅａｋ　ｉｎ　10」は、行番号10で実行が中断されたことを意味しています。

次に、元金Ｐのデータとして０（ゼロ）を入力したときに実行を終了するプログラムに修正してみましょう。ＩＦ…ＴＨＥＮ文を使います。

ここでは、ＩＦ…ＴＨＥＮ文は次の形をしています。なお、ＩＦやＴＨＥＮは小文字のｉｆやｔｈｅｎでもかまいません。

```
IF△変数名=数値△THEN△文(ステートメント)⏎
```

次のようにキーインしましょう。

```
15△if△p=0△then△end⏎
```

スペースを１つあける

この行番号15で元金Ｐが０（ゼロ）以外なら次の行番号の文を実行し、元金Ｐが０ならＥＮＤ文が実行されます。したがって、行番号60のＥＮＤ文は不要となります。そこで、行番号60のＥＮＤ文をデリート（削除）しましょう。次のようにキーインします。

```
60⏎
```

← 行番号60の文（ステートメント）をデリート（削除）

つまり、ある行番号の文（ステートメント）をデリートするには、その行番号をキーインして⏎キーを押せばよいのです。

ここで、プログラムリストをとれば、次のようになります。

```
list
3 REM *** ｶﾞﾝﾘｺﾞｳｹｲ ***
5 CLS
10 INPUT "ｶﾞﾝｷﾝ =";P
15 IF P=0 THEN END
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
55 GOTO 10
Ok
```

15行 ← IF…THEN文のインサート
Ok ← 行番号60のEND文のデリート

それでは、このプログラムを実行してみましょう。

```
ｶﾞﾝｷﾝ =? 100
ﾈﾝﾘ(%) =? 6
ﾈﾝｽｳ =? 10
ｶﾞﾝﾘｺﾞｳｹｲ = 179.085
ｶﾞﾝｷﾝ =? 100
ﾈﾝﾘ(%) =? 7
ﾈﾝｽｳ =? 10
ｶﾞﾝﾘｺﾞｳｹｲ = 196.715
ｶﾞﾝｷﾝ =? 0
Ok
```

ｶﾞﾝｷﾝ =? 0 ← 元金が０（ゼロ）なら
Ok ← 終わり

この実行結果からわかるように、元金Ｐとして０（ゼロ）以外の数値をキーインすれば、元利合計の計算を続けます。元金Ｐとして０をキーインすれば、実行が終了しますね。

空白行を出力するには

この実行結果を見ると、元利合計の値の表示と次の元金データのインプット表示の間に１行間をあけた方が見やすいですね。そこで、ＰＲＩＮＴ文を使って、１行あけることにしましょう。次のようにキーインすれば、１行空白行を出力します。

PRINT ⏎ ← １行空白行を出力

次のようにキーインすればよいわけです。

53△ (SHIFT) ＋ (f･3) ⏎

しかし、ここでは、次のようにしてみましょう。

まず、ＬＩＳＴコマンドを使って、行番号50を画面に表示します。

ＬＩＳＴコマンド

ある行番号の文（ステートメント）を画面に表示するには、ＬＩＳＴコマンドを使って、次のようにキーインします。

LIST △行番号 ⏎

そこで、行番号50を画面に表示するには、次のようにキーインします。

f・4 △ 50 ⏎

```
list 50
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S
Ok
■
```

カーソルはここでブリンク

次に、カーソル移動キー↑、→を使って、カーソルを行番号50のSの右横に動かし、次のようにキーインしてみましょう。

: SHIFT + f・3 ⏎

コロン

プログラムリストをとれば、次のようになります。

```
list
3 REM *** ｶﾞﾝﾘｺﾞｳｹｲ ***
5 CLS
10 INPUT "ｶﾞﾝｷﾝ =";P
15 IF P=0 THEN END
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S:PRINT
55 GOTO 10
Ok
```

マルチステートメント

マルチステートメント

行番号50のように、コロン（:）を使って文と文をつないだ文をマルチステートメントといいます。

実行の結果は、次のとおりです。

```
ｶﾞﾝｷﾝ =? 100
ﾈﾝﾘ(%) =? 6
ﾈﾝｽｳ =? 10
ｶﾞﾝﾘｺﾞｳｹｲ = 179.085

ｶﾞﾝｷﾝ =? 100
ﾈﾝﾘ(%) =? 7
ﾈﾝｽｳ =? 10
ｶﾞﾝﾘｺﾞｳｹｲ = 196.715

ｶﾞﾝｷﾝ =? 0
Ok
```

1行空白行を出力

プログラムに注釈を入れるもう一つの方法

プログラムに注釈（コメント）を入れるには、ＲＥＭ文を使えばよかったですね。

ＲＥＭのかわりに’（アポストロフィ）を使うことができます。

’（アポストロフィ）

「’」と「，（コンマ）」は、まちがいやすいので注意してください。「’」は、キーボードの上のほうの数字７（テンキーの７ではありません）と同じキーです。(SHIFT)キーを押しながらキーを押してください。

ここで、次のようにキーインしてみましょう。

```
3 ' *** ganrigoukei ***
```

小文字

プログラムリストをとれば、次のとおりです。

```
list
3 ' *** ganrigoukei ***
5 CLS
10 INPUT "ｶﾞﾝｷﾝ =";P
15 IF P=0 THEN END
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S:PRINT
55 GOTO 10
Ok
```

注釈文

小文字のまま

行番号３が、新しい注釈文（コメント文）ですね。

このように、すでにある行番号と同じ行番号の文をキーインすると、すでにある行番号は消えて、新しくキーインした文に置きかわります。

また、プログラムリストを見ればわかるように、注釈文の中のアルファベットは小文字のままで大文字にかわっていませんね。ＰＣ-８８０１は、アルファベットを小文字でキーインしても、次の場合を除いて、大文字に変換します。

❶ 注釈文の中の注釈としてのアルファベット

❷ ″（ダブルクォーテーションマーク）で囲まれた中のアルファベット

そこで、コマンド、文（ステートメント）、関数など、アルファベットをキーインするときは、上の２つの場合を除いて、大文字でも小文字でもかまわないことになります。

なお、注釈文として、ＲＥＭを使うかアポストロフィ（’）を使うかは好みの問題です。好きなほうを使いましょう。

行番号３の注釈文を自分の好みの注釈文にかえてみましょう。

ここでは、次のようにキーインしておきます。

```
3 rem *** ｶﾞﾝﾘｺﾞｳｹｲ ***
```

プログラムリストは、次のとおりです。

```
list
3 REM *** ｶﾞﾝﾘｺﾞｳｹｲ ***     ←―――― 注釈文
5 CLS
10 INPUT "ｶﾞﾝｷﾝ =";P
15 IF P=0 THEN END
20 INPUT "ﾈﾝﾘ(%) =";R
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S:PRINT
55 GOTO 10
Ok
```

文字の色を変えてみよう

ＰＣ-８８０１は、文字の色として８色のカラーを表示することができます。もちろん、ディスプレイがカラーディスプレイでなければ、カラー表示はできません。

ＣＯＮＳＯＬＥ文

画面を白黒モードからカラーモードに変えるためには、ＣＯＮＳＯＬＥ（コンソール）文を使います。次のようにキーインしてください。

```
7△console△,,,1⏎   ←―――― カラーモードの指定
                       スペースを１つあける
```

ＣＯＬＯＲ文

次に、ＣＯＬＯＲ文を使って、文字の色を指定します。ＣＯＬＯＲ文と文字の色との対応は、次のとおりです。

ＣＯＬＯＲ文	文字の色
ＣＯＬＯＲ△０	黒
ＣＯＬＯＲ△１	青
ＣＯＬＯＲ△２	赤
ＣＯＬＯＲ△３	紫
ＣＯＬＯＲ△４	緑
ＣＯＬＯＲ△５	水色
ＣＯＬＯＲ△６	黄
ＣＯＬＯＲ△７	白

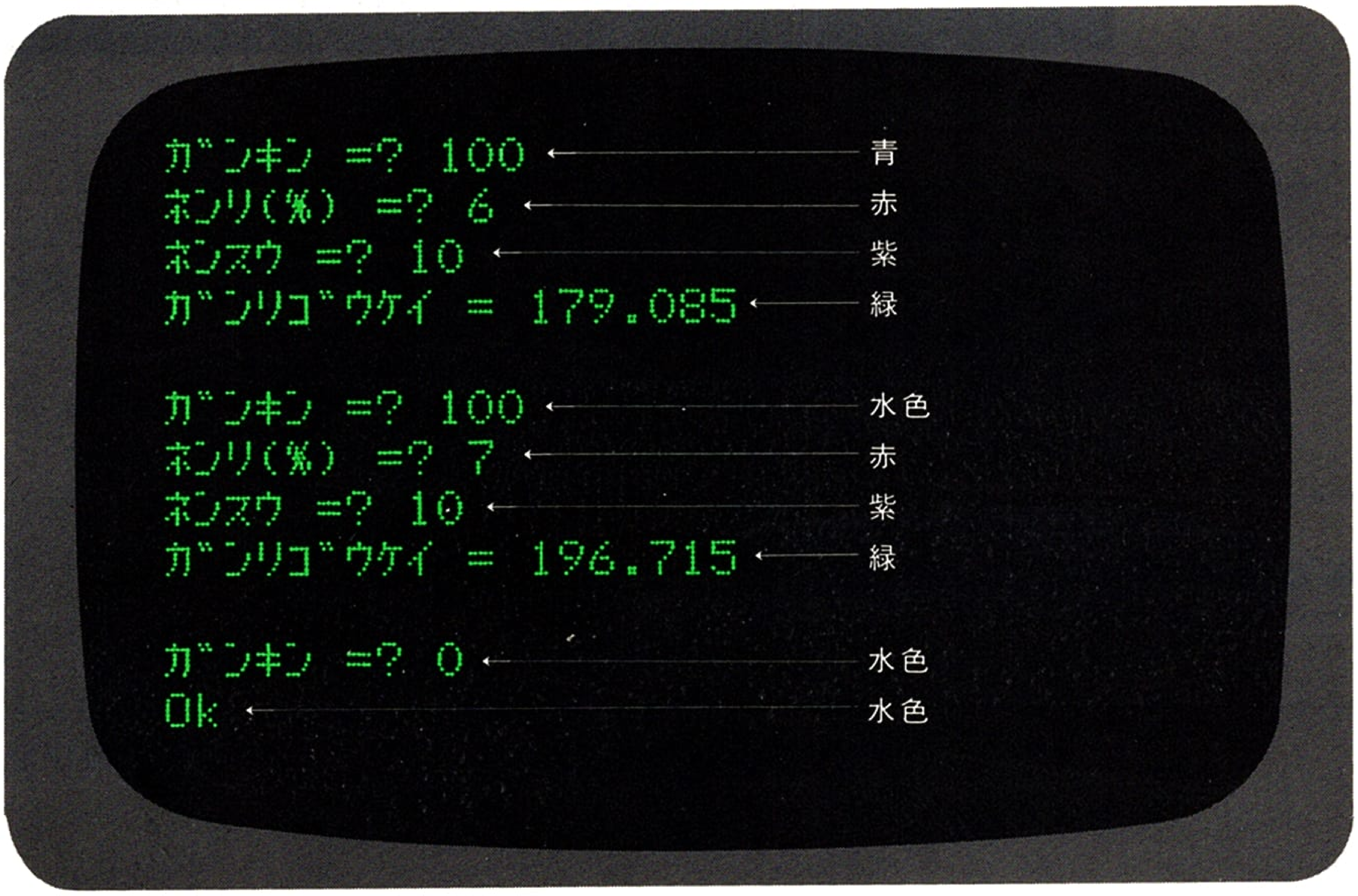
それでは、ＣＯＬＯＲ文を使って、文字の色を変えてみましょう。次のようにキーインしてみます。

```
8 COLOR 1 ↵
18 COLOR 2 ↵
28 COLOR 3 ↵
48 COLOR 4 ↵
53 COLOR 5 ↵
```

プログラムリストは、次のとおりです。

```
list
3 REM *** ｶﾞﾝﾘｺﾞｳｹｲ ***
5 CLS
7 CONSOLE ,,,1
8 COLOR 1
10 INPUT "ｶﾞﾝｷﾝ =";P
15 IF P=0 THEN END
18 COLOR 2
20 INPUT "ﾈﾝﾘ(%) =";R
28 COLOR 3
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
48 COLOR 4
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S:PRINT
53 COLOR 5
55 GOTO 10
Ok
```

実行してみましょう
画面の文字の色は、次のように変わります。



しかし、このままにしておくと、画面の色はいつまでも水色のままです。そこで、元金Ｐとして０（ゼロ）をキーインしたら、画面をカラーモードから白黒モードにもどすことにしましょう。

まず、元金Ｐとして０（ゼロ）をキーインしたら、行番号60へ飛ぶことにします。ＩＦ…ＧＯＴＯ文を使います。ここでは、ＩＦ…ＧＯＴＯ文は、次の形をしています。

```
IF △ 変数名＝数値 △ GOTO △ 行番号 ⏎
```

それでは、ＬＩＳＴコマンドを使って行番号15を画面に表示します。

```
list △ 15 ⏎
```



次に、行番号15を次のように修正します。

まず、カーソル移動キー↑、→を使って、カーソルを動かし、行番号15のＴの上で止めます。



そして、次のようにキーインします。

```
f·3 60 ⏎
```

これで、行番号15で元金Ｐが０（ゼロ）以外なら次の行番号の文を実行し、元金Ｐが０なら行番号60へ飛びます。

そこで、行番号60でカラーモードから白黒モードにもどします。ＣＯＮＳＯＬＥ文を使って、次のようにキーインしてください。

```
60 △ console △ ,,,0 ⏎
```

白黒モードの指定

スペースを１つあける

そして、行番号70でＥＮＤ文を使って、プログラムの実行を終了させます。

```
70 △ end ⏎
```
← プログラムの実行を終了させる

プログラムリストは、次のとおりです。

```
list
3 REM *** ｶﾞﾝﾘｺﾞｳｹｲ ***
5 CLS
7 CONSOLE ,,,1
8 COLOR 1
10 INPUT "ｶﾞﾝｷﾝ =";P
15 IF P=0 GOTO 60
18 COLOR 2
20 INPUT "ﾈﾝﾘ(%) =";R
28 COLOR 3
30 INPUT "ﾈﾝｽｳ =";N
40 S=P*(1+R/100)^N
48 COLOR 4
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S:PRINT
53 COLOR 5
55 GOTO 10
60 CONSOLE ,,,0
70 END
Ok
```

実行してみましょう。

```
ｶﾞﾝｷﾝ =? 100
ﾈﾝﾘ(%) =? 6
ﾈﾝｽｳ =? 10
ｶﾞﾝﾘｺﾞｳｹｲ = 179.085

ｶﾞﾝｷﾝ =? 100
ﾈﾝﾘ(%) =? 7
ﾈﾝｽｳ =? 10
ｶﾞﾝﾘｺﾞｳｹｲ = 196.715

ｶﾞﾝｷﾝ =? ■
```
← 元金はいくらですか？

ここで、元金Ｐのデータとして０（ゼロ）をキーインすると、画面はクリアされ、白黒モードになります。画面左上に白黒でＯｋが表示されますね。



プログラムの行番号を整理する

RENUMコマンド

ここで、ＲＥＮＵＭ（リナム）コマンドを使って、プログラムの行番号を整理しておきましょう。

プログラムの行番号を10からはじまって、10、20、30、……のように10きざみに行番号をつけかえるには、次のようにキーインします。

```
renum ↩
```

← 10、20、30、……と行番号をつけかえる

プログラムリストをとれば、次のとおりです。

```
list
10 REM *** ｶﾞﾝﾘｺﾞｳｹｲ ***
20 CLS
30 CONSOLE ,,,1
40 COLOR 1
50 INPUT "ｶﾞﾝｷﾝ =";P
60 IF P=0 GOTO 160
70 COLOR 2
80 INPUT "ﾈﾝﾘ(%) =";R
90 COLOR 3
100 INPUT "ﾈﾝｽｳ =";N
110 S=P*(1+R/100)^N
120 COLOR 4
130 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ =";S:PRINT
140 COLOR 5
150 GOTO 50
160 CONSOLE ,,,0
170 END
Ok
```

行番号が10，20，30，……につけかえられた

最後に、この元利合計プログラムをセーブしておきましょう。

4·11 まとめ

コマンドステートメント（文）

コマンドとステートメント（文）は、ＢＡＳＩＣの命令です。コマンドは、一般に、ダイレクトモードで使用するもので、実行後Ｏｋが表示されます。ステートメント（文）は、主に、プログラムモードで使用します。しかし、コマンドとステートメント（文）は、明確に分かれているわけではありません。

コマンド
ステートメント(文)

なお、コマンドやステートメント（文）をキーインするとき、アルファベットは、次の２つの場合を除いて、大文字でも小文字でもかまいません。

(1)　ＲＥＭ文の注釈としてのアルファベット
(2)　〝（ダブルクォーテーションマーク）で囲まれた中のアルファベット

ただし、コマンドやステートメントのキーワードの後には、必ず、スペースを入れなければなりません。

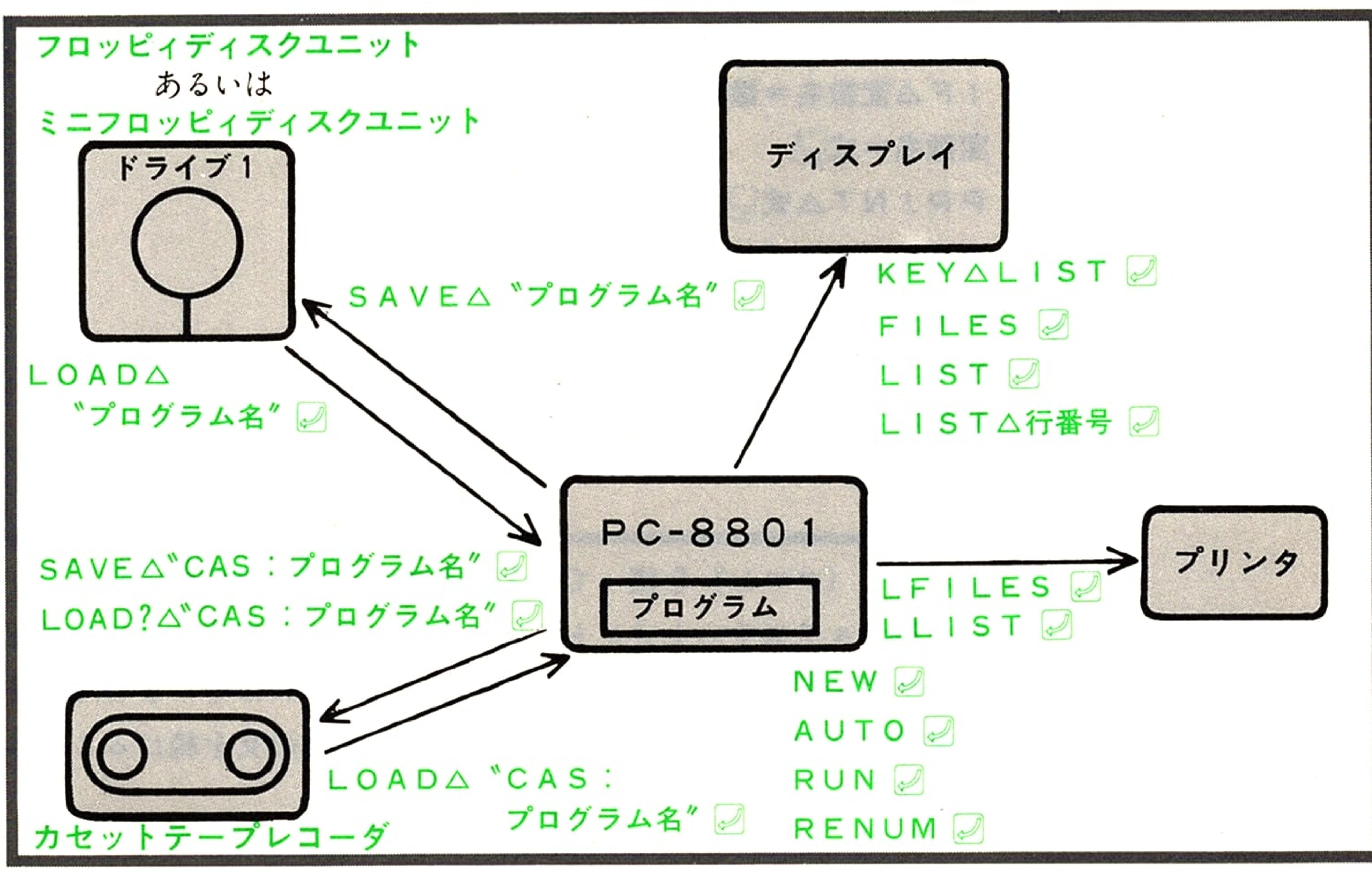
スペース

本書に出てくる主なコマンドとステートメント（文）は、表紙および裏表紙の見返しにまとめてあります。参考にしてください。

コマンドのまとめ

いままでに出てきた主なコマンドをまとめておきましょう。

コマンド



ステートメント（文）のまとめ

ダイレクトモードの行は、次の形になっています。

　　ステートメント（文）↵

プログラムモードの行は、次の形になっています。

　　行番号△ステートメント（文）↵

ステートメント

いままでに出てきたステートメントをまとめておきましょう。

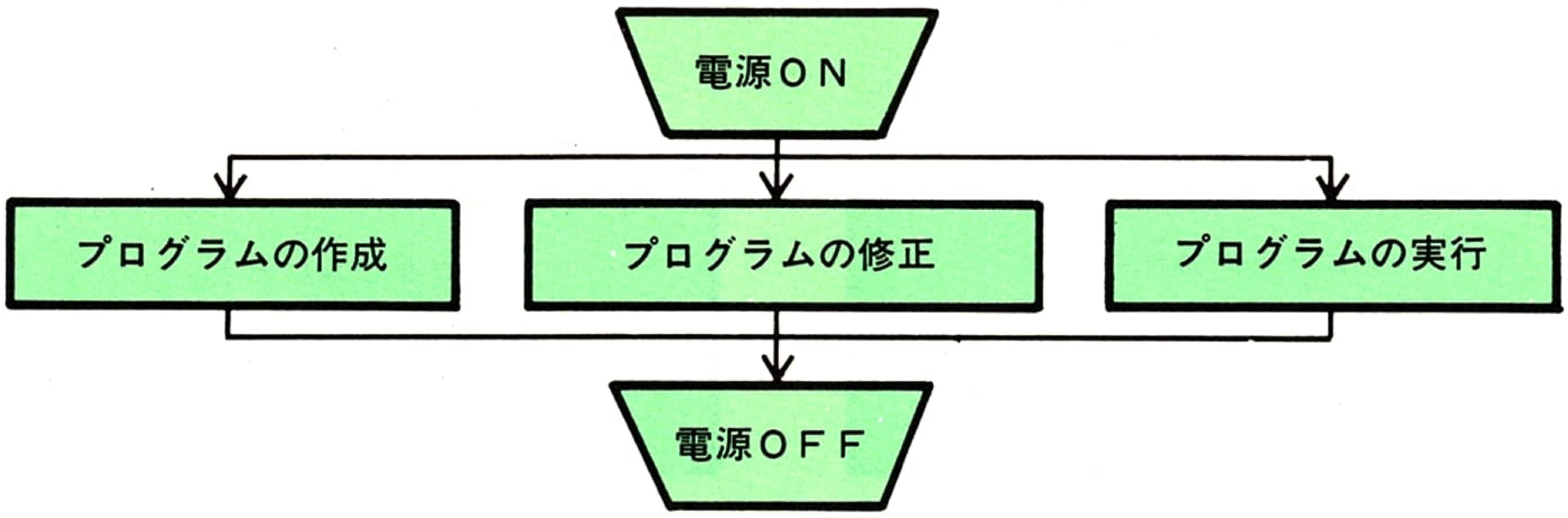
```
KEY△キー番号，“定義内容”↵
MOTOR↵
REM△注釈↵
CLS↵                    ← 画面クリア
CONSOLE△，，，1↵        ← カラーモード
COLOR△0↵                ← 黒    ┐
COLOR△1↵                ← 青    │
COLOR△2↵                ← 赤    │
COLOR△3↵                ← 紫    │← カラーモードの場合
COLOR△4↵                ← 緑    │
COLOR△5↵                ← 水色  │
COLOR△6↵                ← 黄色  │
COLOR△7↵                ← 白    ┘
INPUT△“メッセージ”；変数名↵
IF△変数名＝数値△THEN△ステートメント（文）↵
IF△変数名＝数値△GOTO△行番号↵
変数名＝式↵
PRINT△式↵
PRINT△“メッセージ”；変数名↵
PRINT↵
GOTO△行番号↵
CONSOLE△，，，0↵        ← 白黒モード
END↵
```

マルチステートメント

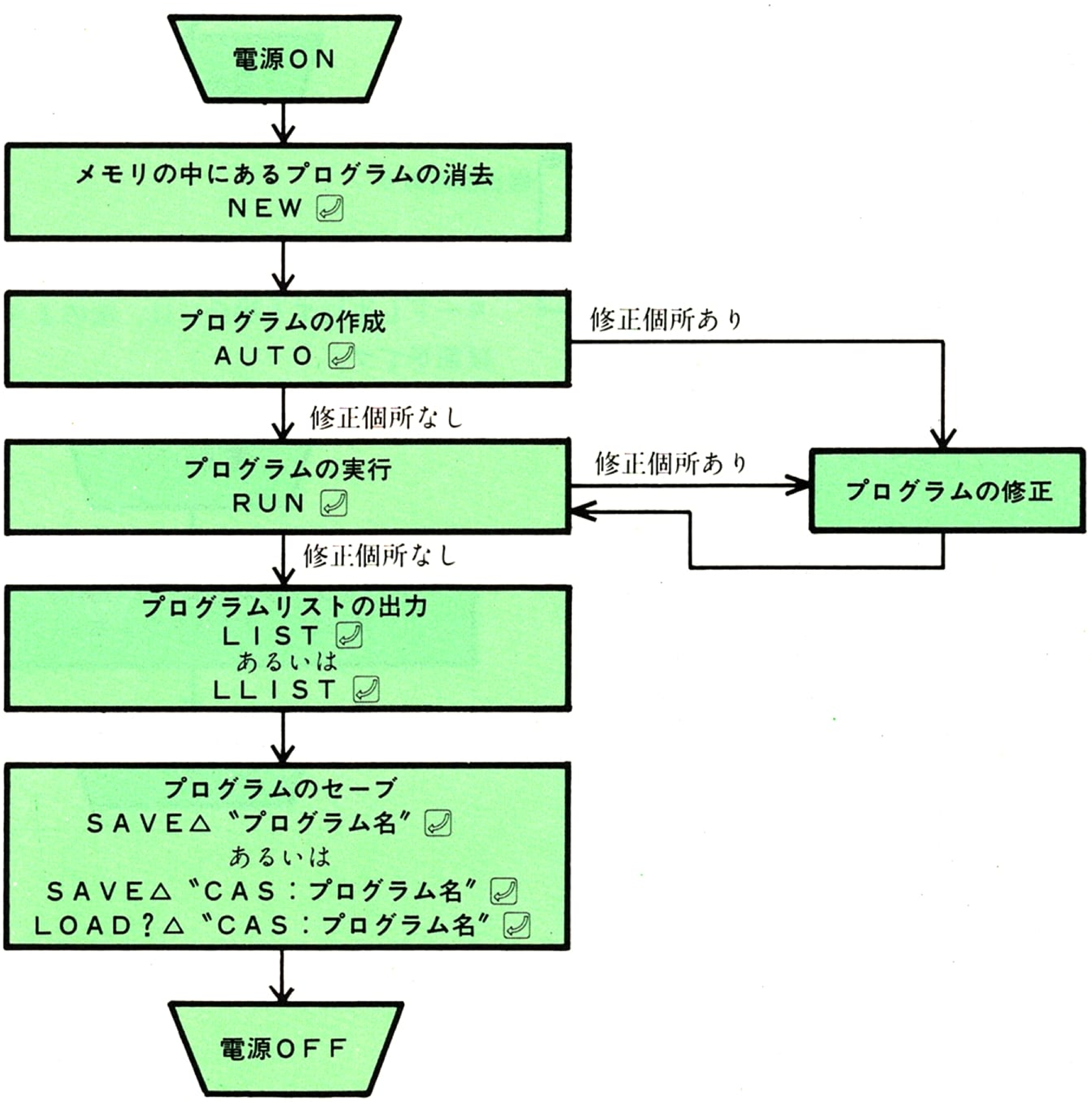
なお、：（コロン）を使って文と文をつなぐことができます。こうした文をマルチステートメントといいます。マルチステートメントは行番号を含めて、1行 255文字以内の長さでなければなりません。なお、REM文の行は、：（コロン）で区切って他の文を続けることはできません。

基本操作のまとめ

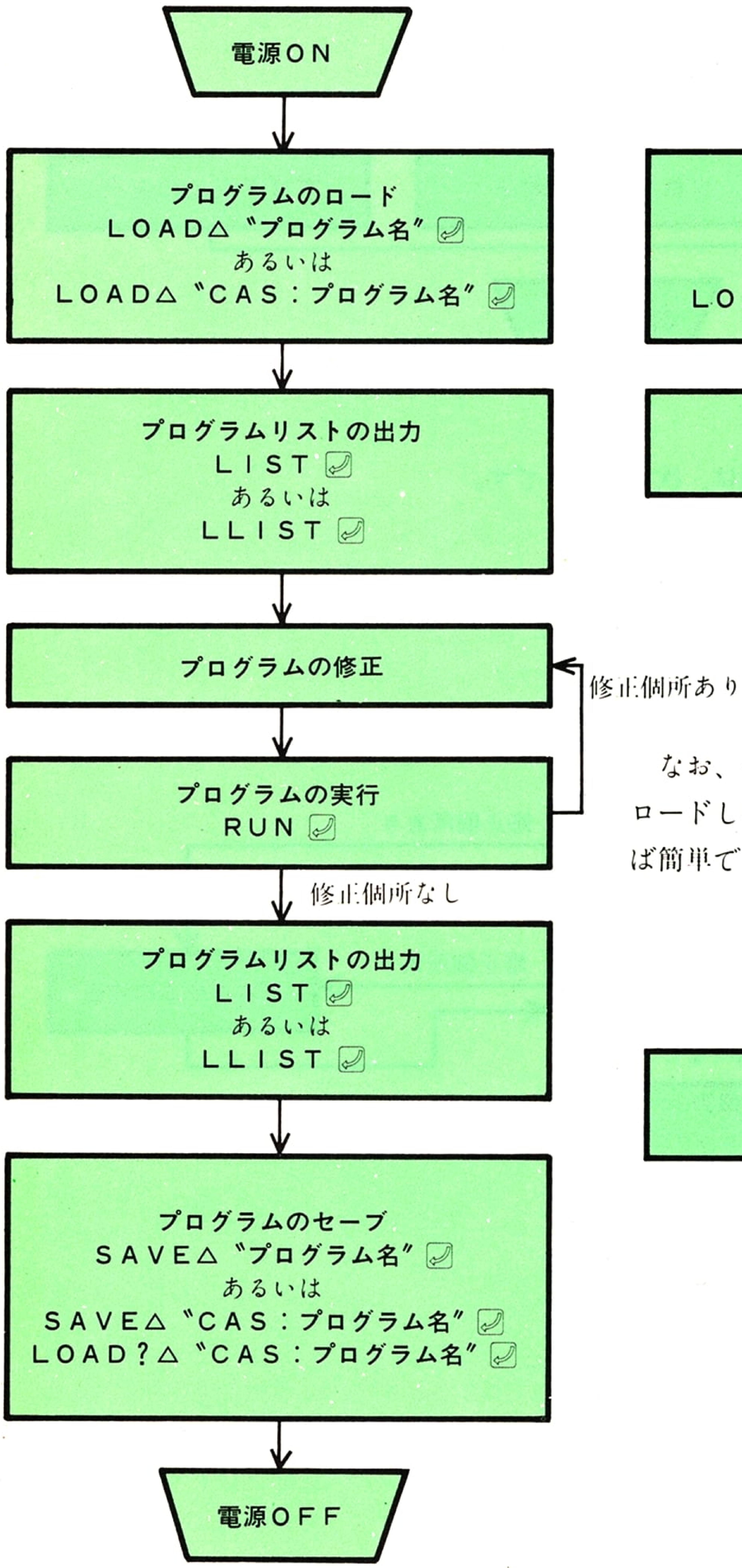
ここで、ＰＣ-８８０１の基本操作をまとめておきましょう。



新しくプログラムを作成する手順は、次のとおりです。



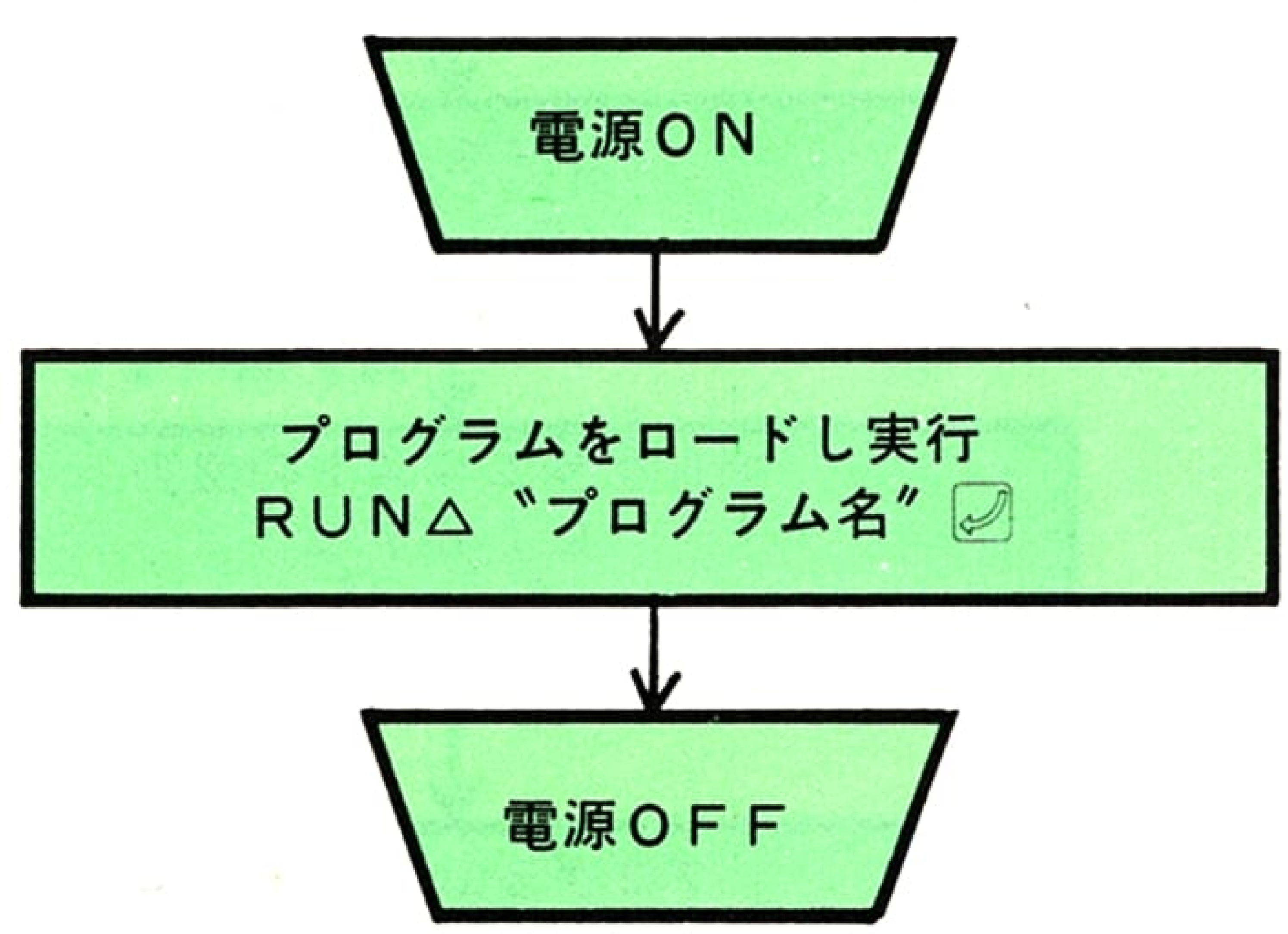
既存のプログラムを修正する手順は、次のとおりです。



既存のプログラムを実行するだけの場合は、次のようにします。



なお、ディスクからプログラムをメモリにロードし実行する場合には、次のようにすれば簡単です。



第5章

経営分析プログラム

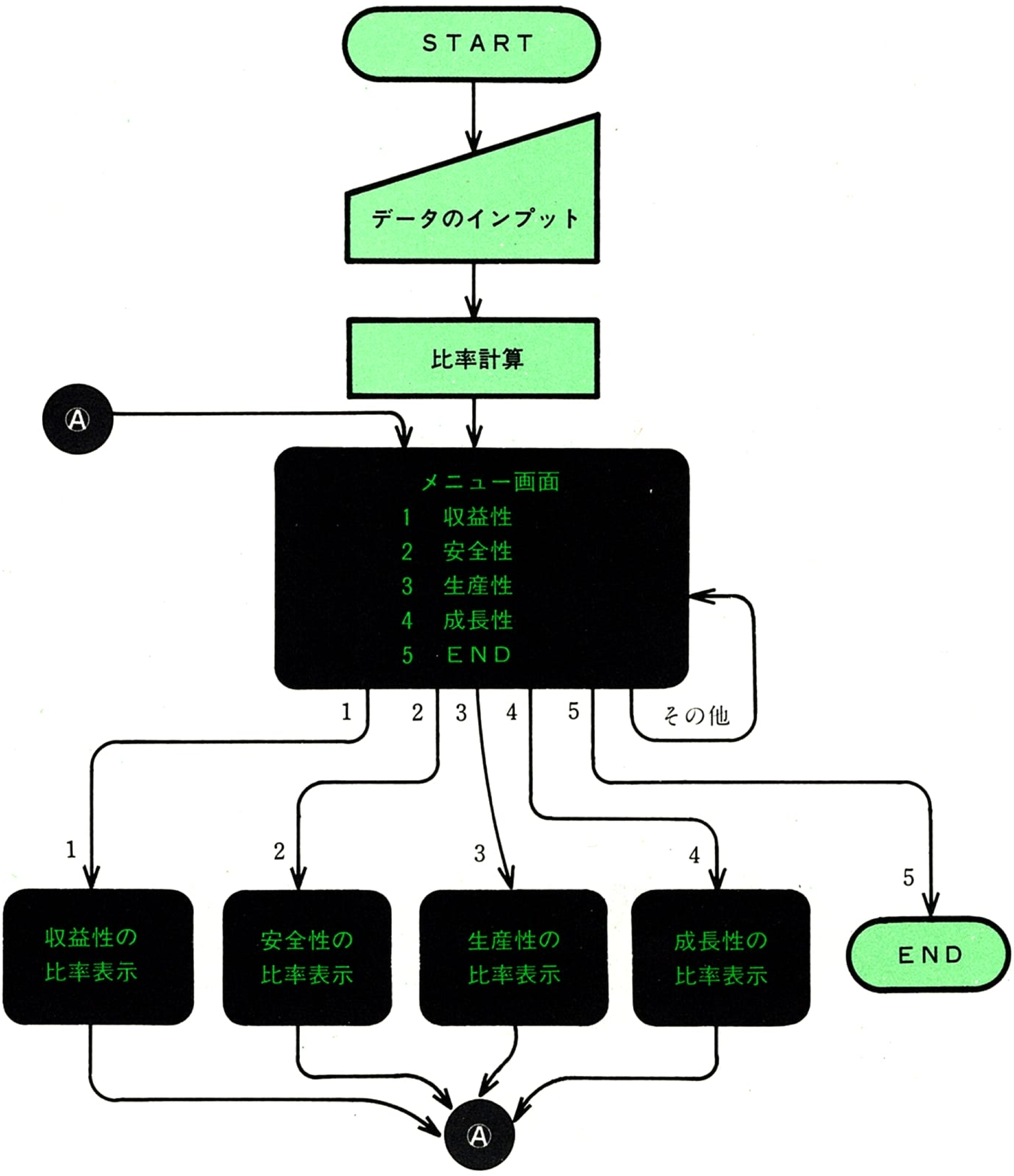
5・1　経営分析プログラムのフローチャート

経営分析とは何か

貸借対照表（B／S，ビーエス）や損益計算書（P／L、ピーエル）などの数値にもとづいて、企業の収益性、安全性、生産性、成長性などを分析する手法を経営分析といいます。

経営分析プログラムのフローチャート

ここで作成する経営分析プログラムのフローチャート(流れ図）は、次のとおりです。



次に、このフローチャートにしたがって、経営分析プログラムの全体的な流れを見てみましょう。

まず、経営分析プログラムをスタートさせます。

そして、データをインプットします。ここでは、インプット項目は次の12項目です。

(1) 流動資産
(2) 固定資産
(3) 流動負債
(4) 自己資本
(5) 期首総資本
(6) 期末総資本

← 貸借対照表（B／S）の項目

(7) 前期売上高
(8) 当期売上高
(9) 前期経常利益
(10) 当期経常利益

← 損益計算書（P／L）の項目

(11) 期首従業員数
(12) 期末従業員数

インプットデータにもとづいて、比率計算をします。ここでは、次の10比率を計算します。

(1) 収益性

❶ 平均総資本経常利益率（%）

$$= \frac{\text{経常利益}}{(\text{期首総資本}+\text{期末総資本})\div 2} \times 100$$

❷ 売上高経常利益率（%）

$$= \frac{\text{経常利益}}{\text{売上高}} \times 100$$

❸ 平均総資本回転率（回）

$$= \frac{\text{売上高}}{(\text{期首総資本}+\text{期末総資本})\div 2}$$

(2) 安全性

❶ 流動比率（%）

$$= \frac{\text{流動資産}}{\text{流動負債}} \times 100$$

❷ 固定比率（%）

$$= \frac{\text{固定資産}}{\text{自己資本}} \times 100$$

❸ 自己資本比率（%）

$$= \frac{\text{自己資本}}{\text{総資本}} \times 100$$

(3) 生産性

❶ 従業員1人あたり売上高

$$= \frac{\text{売上高}}{(\text{期首従業員数}+\text{期末従業員数})\div 2}$$

❷ 従業員1人あたり経常利益

$$= \frac{\text{経常利益}}{(\text{期首従業員数}+\text{期末従業員数})\div 2}$$

(4) 成長性

❶ 増収率（%）

$$= \frac{\text{当期売上高}-\text{前期売上高}}{\text{前期売上高}}\times 100$$

❷ 経常増益率（%）

$$= \frac{\text{当期経常利益}-\text{前期経常利益}}{\text{前期経常利益}}\times 100$$

以上の10比率のうち、固定比率だけは、その値が小さいほど望ましい。この点、注意が必要です。

続いて、ディスプレイに、次のメニュー画面が表示されます。



ここで、1をキーインすれば収益性の比率表示、2をキーインすれば安全性の比率表示、3をキーインすれば生産性の比率表示、4をキーインすれば成長性の比率表示をすることができます。5をキーインすれば経営分析プログラムの実行を終了します。

収益性、安全性、生産性、成長性のうちの1つの表示がすめば、メニュー画面にもどります。

それでは、このフローチャートにしたがって、経営分析プログラムを作成してみましょう。

なお、貸借対照表（B／S）や損益計算書（P／L）の見方、経営分析比率の意味などについては、たとえば、次のものを参照してください。

- 日本経済新聞社編『財務諸表の見方』日経文庫(6)、日本経済新聞社、昭和51年
- 青木茂男著『経営分析の手引』日経文庫(151)、日本経済新聞社 昭和55年。

5・2 スタート部分のプログラム

プログラムのキーインの前に

まず、NEWコマンドを使って、PC−8801のメモリにあるプログラムを消去します。　NEWコマンド

```
NEW ⏎  ←――― メモリにあるプログラムを消去
```

行番号を自動発生させる

次に、AUTOコマンドを使って、行番号を10から始まって10きざみで発生させることにします。次のようにキーインすればよかったですね。　AUTOコマンド

```
(f・2) ⏎  ←――― 行番号を10からはじまって10きざみで発生させる
```

プログラムに注釈を入れる

REM文を使って、プログラムに注釈（コメント）を入れます。　REM文

```
10 REM ***** ｹｲｴｲ ﾌﾞﾝｾｷ ***** ⏎  ←――― 注釈（コメント）文
```

配列の宣言

いろいろな値をとることができる数を変数といい、その名前を変数名といいましたね。なお、変数名は、最大40文字の英字で始まる英数字とピリオド（.）で表されます。　変数　変数名

いま、変数X、Y、Zがあるとき、これら3つの変数を1つの変数名A(1)、A(2)、A(3)で取り扱うことができます。このように、1つの変数名でいくつかの要素を参照することのできる変数を配列変数といいます。　配列変数

変　数		配列変数
X	――→	A(1)
Y	――→	A(2)
Z	――→	A(3)

配列変数名は、普通の変数名の後ろに（　）をつけます。そして、（　）の中を配列変数の添字（そえじ）といいます。　配列変数名　添字

配列変数の大きさは、DIM（ディム）文によって宣言します。DIM文は、次の形をしています。なお、DIMはdimension（ディメンジョン）の略で、大きさという意味です。　DIM文

DIM △配列変数名（添字の最大値）、配列変数名（添字の最大値）、…… ←DIM文

ＤＩＭ文の配列変数の添字には、その最大値を入れます。また、ＤＩＭ文には、配列変数名をコンマ（，）で区切って並べて書くことができます。

文字列
ストリング

ところで、英数字、カナ文字、記号などの文字の並んだもの（文字の列）のことを文字列あるいはストリングといいます。 ストリング（文字列）を入力したり、記憶させたり、出力したりするときに、ストリングを変数（ストリング変数）として扱うと便利です。変数名、配列変数名の最後に＄（ダラー）をつければ、文字型変数、文字型配列変数となります。

文字型変数
文字型配列変数

それでは、次のようにキーインしましょう。

```
20 DIM A$(12),B(12),C$(10),D(10)
```
←配列の宣言

ここで、Ａ＄(12)は、12のインプット項目名の文字型配列変数です。Ｂ（12）は、12のインプットデータの配列変数です。次に、Ｃ＄(10)は、10の比率名の文字型配列変数です。Ｄ（10）は、計算される10の比率の配列変数です。

なお、配列変数の添字は０から始まりますから、実際の配列変数の要素の数は添字の数＋１となります。

	Ａ＄(12)	Ｂ(12)	Ｃ＄(10)	Ｄ(10)
0	Ａ＄(０)	Ｂ(０)	Ｃ＄(０)	Ｄ(０)
1	Ａ＄(１)	Ｂ(１)	Ｃ＄(１)	Ｄ(１)
2	Ａ＄(２)	Ｂ(２)	Ｃ＄(２)	Ｄ(２)
⋮	⋮	⋮	⋮	⋮
10	Ａ＄(10)	Ｂ(10)	Ｃ＄(10)	Ｄ(10)
11	Ａ＄(11)	Ｂ(11)		
12	Ａ＄(12)	Ｂ(12)		

また、ＤＩＭ文が実行された時点で、配列変数のすべての要素〔たとえば、Ｂ（０）…Ｂ（12）、Ｄ（０）…Ｄ（10）〕の値は０に設定され文字型配列変数のすべての要素〔たとえば、Ａ＄（０）…Ａ＄（12）、Ｃ＄（０）…Ｃ＄（10）〕にはヌルストリング（空の文字列）が設定されます。

画面のクリア

ＣＬＳ文

次に、ＣＬＳ文を使って、画面をクリアします。

```
30 CLS
```
←画面のクリア

画面の桁数、行数を指定

画面の桁数、行数を指定するには、WIDTH（ワイズ）コマンドを使います。WIDTHコマンドには、次の4種類があります。

WIDTHコマンド

WIDTH△40，20 ⏎ ……画面を40桁20行にします
WIDTH△40，25 ⏎ ……画面を40桁25行にします
WIDTH△80，20 ⏎ ……画面を80桁20行にします
WIDTH△80，25 ⏎ ……画面を80桁25行にします

ここでは、次のようにキーインしておきましょう。

```
40 WIDTH 40,25 ⏎
```
← 画面を40桁25行にする

画面のモード設定を行なう

次に、CONSOLE（コンソール）文を使って、画面のモード設定を行ないます。CONSOLE文は、次の形をしています。

CONSOLE文

CONSOLE △スクロール開始行、スクロール行数、ファンクションキー表示スイッチ、カラー／白黒スイッチ ⏎ ← CONSOLE文

スクロール開始行とスクロール行数を指定することによって、画面上でスクロールする領域（スクロールウィンドウ）を指定します。

ファンクションキー表示スイッチに1を指定すると、画面最下位行にファンクションキーに定義されているストリング（文字列）を表示します。0を指定すれば、ファンクションキーの表示を消すことができます。

カラー／白黒スイッチに1を指定すると、画面をカラーモードにします。0を指定すれば、白黒モードになります。

なお、これら4つのパラメータのうち、その値を変更しないものは省略できます。ただし、コンマ（，）は省略できません。

ここでは、次のようにキーインします。

```
50 CONSOLE 0,25,0,0 ⏎
```
← ファンクションキーの表示を消し、白黒モードにする

画面の0行め（最上位行）から25行をスクロールさせます。行番号40のWIDTHコマンドで画面の行数が25行に指定してありますから、最下位行まで文字を表示し、さらに次の行の文字を表示させると、最上位の行を消し、画面全体が1行上へ移動します。この動作をスクロールといいます。

また、ファンクションキーの表示は消し、白黒モードになります。

なお、文字の色を変えてみたい人は、ここでカラーモードに指定し

ておき、以後ＣＯＬＯＲ文を使ってみてください。

見出しの表示

次のようにキーインして、画面に見出しを表示し、その後１行空白行を出力しておきます。

```
60 PRINT "********* ｹｲｴｲ ﾌﾞﾝｾｷ *********" ⏎ ←見出しの表示
70 PRINT ⏎ ←―――――――――― 1行空白行を出力
```

行番号の自動発生をストップさせる

ここで、(STOP)キーを押して、行番号の自動発生をストップさせます。

経営分析プログラムのスタート部分のリスト

(CLR)キーを押して、画面をクリアします。次に、(f･4)キーを押して、経営分析プログラムのスタート部分のリストをとってみましょう。次のようになりましたか？

```
10 REM ***** ｹｲｴｲ ﾌﾞﾝｾｷ *****
20 DIM A$(12),B(12),C$(10),D(10)
30 CLS
40 WIDTH 40,25
50 CONSOLE 0,25,0,0
60 PRINT "********* ｹｲｴｲ ﾌﾞﾝｾｷ *********"
70 PRINT
```
→経営分析プログラムのスタート部分

修正個所があれば、カーソル移動キー（↑、←、→、↓）、(DEL)キー、(INS)キーを使って、修正しましょう。

5・3 データインプットのプログラム

注釈を入れる

データインプットのプログラムは、経営分析プログラムのスタート部分とは区別して、行番号100から始めることにします。

行番号を100から10きざみに、100、110、120、……と自動的に発生させるためには、AUTOコマンドを使って、次のようにします。　**AUTOコマンド**

```
f・2  100 ⏎   ← 行番号を100、110、120、……と自動的に発生させる
```

行番号100にREM文を使って、注釈(コメント)を入れます。　**REM文**

```
100 REM ***** DATA / INPUT ***** ⏎   ← 注釈(コメント)文
```

インプット項目名を文字型配列変数に代入する

ここで、次の代入文を使って、12のインプット項目名を文字型配列変数に代入します。　**代入文**

```
文字型配列変数＝"文字列" ⏎   ← 文字列を文字型配列変数に代入
```

次のとおりです。

```
110 A$(1)="リュウドウシサン                :
120 A$(2)="コテイシサン                    :
130 A$(3)="リュウドウフサイ                :
140 A$(4)="ジコシホン                      :
150 A$(5)="キシュ ソウシホン               :
160 A$(6)="キマツ ソウシホン               :
170 A$(7)="ゼンキ ウリアゲダカ             :
180 A$(8)="トウキ  ウリアゲダカ            :
190 A$(9)="ゼンキ ケイジョウリエキ         :
200 A$(10)="トウキ  ケイジョウリエキ        :
210 A$(11)="キシュ ジュウギョウインスウ     :
220 A$(12)="キマツ ジュウギョウインスウ     :
```

← インプット項目を文字型配列変数に代入

これらの代入文の右辺は、すべて、文字定数(リテラル)と呼ばれます。文字定数とは、255文字以下のダブルクォーテーション(")で囲まれた英数字、カナ文字、記号などのストリング(文字列)のことです。　**文字定数**

また、代入文の左辺の文字型配列変数は、行番号20のDIM文で配列の宣言がなされていましたね。

これらの代入文を実行すると、右辺のストリング（文字列）が左辺の文字型配列変数に代入されます。

なお、同じような文字列が出てきますから、次のようにして、キーインを簡単にすることができます。

たとえば、行番号150の代入文をキーインしたら、[↑]キーを押し、行番号150にもどります。150を160に変え、Ａ＄（５）の５を６に変え、さらに、キシュをキマツに変え、[↵]キーを押します。これで、行番号160の代入文のキーインができたことになります。

このように、同じような文字列が出てきたら、キーインをくふうしてみましょう。

データインプット

データインプットのプログラムは、次のとおりです。

```
230 PRINT "--- data ｦ input ｼﾃｸﾀﾞｻｲ ---"
240 PRINT
250 FOR I=1 TO 12             ─┐
260 PRINT "(";                 │
270 PRINT USING "##";I;  ← *   │
280 PRINT ") ";A$(I);          │ ← FOR～NEXT文
290 INPUT B(I)           ← **  │
300 NEXT I                    ─┘
          *  PRINT USING文
          ** INPUT文
```

行番号230でメッセージを画面に表示し、行番号240で１行空白行を出力します。

行番号250のＦＯＲ文と行番号300のＮＥＸＴ文は、次のように対(つい)あるいはペァになっていて、ＦＯＲ～ＮＥＸＴ文といいます。

ＦＯＲ～ＮＥＸＴ文

```
FOR △ 変数名＝初期値△ＴＯ△最終値 [↵] ←─┐
 ∫                                          ├── FOR～NEXT文
NEXT △ 変数名 [↵] ←─────────────────┘
```

プログラムの実行がＦＯＲ文までくると、変数に初期値が代入されます。変数の値が最終値と等しいか小さければ、ＦＯＲ文とＮＥＸＴ文の間の文を実行します。ＮＥＸＴ文になると、変数に１が足されます。変数値が最終値と等しいか小さいかぎり、ＦＯＲとＮＥＸＴの間の文を繰り返し実行します。

つまり、ＦＯＲ～ＮＥＸＴ文は、ＦＯＲ文とＮＥＸＴ文の間の文を繰り返し実行させるためのものです。繰り返しのことをループといいますので、ＦＯＲ～ＮＥＸＴ文によって繰り返されるループのことをＦＯＲ～ＮＥＸＴループといいます。

FOR～NEXTループ

そして、変数の値が最終値をこえたときに、ＦＯＲ～ＮＥＸＴループから脱出して、ＮＥＸＴ文の次の文を実行します。

ところで、行番号250と300のＦＯＲ～ＮＥＸＴループのＰＲＩＮＴ文とＩＮＰＵＴ文は、どんな意味を持っているのでしょうか。

PRINT文

行番号260のＰＲＩＮＴ文を実行すると、画面に（を表示します。なお、ＰＲＩＮＴ文の最後がセミコロン（；）で終わっていますね。こうしておけば、改行動作をおこさず、つまり、画面上その行で次のＰＲＩＮＴ文による出力が続けて表示されます。

PRINT USING文

行番号270は、次の形をしたＰＲＩＮＴ　ＵＳＩＮＧ文です。

```
PRINT△USING△"##";変数名 ⏎
```

“＃＃”は、変数の数値を出力する桁数を２桁に指定するものです。なお、指定した桁数より変数の数値の桁数が小さい場合には、右づめで出力します。

行番号270のＰＲＩＮＴ　ＵＳＩＮＧ文を実行すれば、画面に変数Ｉの値を２桁で表示します。そして、行番号270のＰＲＩＮＴ　ＵＳＩＮＧ文も最後がセミコロン（；）で終わっているので、改行動作をおこしません。

次に、行番号280のＰＲＩＮＴ文を実行すれば、画面に）を表示し、配列変数Ａ＄（Ｉ）のストリング（文字列）を表示します。改行動作はおこしません。

INPUT文

そして、行番号290のＩＮＰＵＴ文を実行すれば、インプットデータが、配列変数Ｂ（Ｉ）に代入されます。なお、この配列変数は、行番号20のＤＩＭ文で配列の宣言がなされていましたね。

以上のＦＯＲ～ＮＥＸＴループを、変数Ｉが１から12まで、つまり、12回実行します。12は、インプット項目の数のことですね。

インプットデータの画面表示

インプットデータは画面に表示されますが、右づめで出力されませんし、３桁ごとにコンマ（，）で区切って表示されません。そこで、インプットデータを見やすくするくふうをしてみましょう。

```
310 CLS
320 PRINT "*** input data ***"
330 PRINT
340 FOR I=1 TO 12
350 PRINT "(";
360 PRINT USING "##";I;
370 PRINT ") ";A$(I);
380 PRINT USING "#,###,###";B(I)
390 NEXT I
```

（340〜390：FOR〜NEXT文）

＊ ＰＲＩＮＴ ＵＳＩＮＧ文（行番号360、380）

まず、画面をクリアし、見出しを表示し、１行空白行を出力します。そして、ＦＯＲ〜ＮＥＸＴ文によって、ＦＯＲ〜ＮＥＸＴループを12回繰り返し実行します。

ＦＯＲ〜ＮＥＸＴループの中は、データのインプットとほぼ同じです。ちがうところは、行番号380のＰＲＩＮＴ ＵＳＩＮＧ文だけです。データのインプットの場合は、ここが行番号290のＩＮＰＵＴ文でした。

行番号380のＰＲＩＮＴ ＵＳＩＮＧ文は、インプットデータが代入されている配列変数Ｂ（Ｉ）の値を、右づめで３桁ごとにコンマ（，）で区切って出力させるためのものです。

以上の文のキーインにあたっては、行番号の自動発生をストップさせた上で、次のようにすれば簡単にキーインができます。

ＬＩＳＴコマンド

ＬＩＳＴコマンドを使って、指定した行番号から行番号までの範囲のすべての行を表示します。

LIST △ 行番号 − 行番号 ⏎ ← 指定した行番号から行番号の範囲の行を画面に表示

そこで、次のようにキーインします。

(f·4) 230−300 ⏎ ← 行番号230〜300の範囲の行を画面に表示

行番号310〜390の行の中で、行番号230〜300の行と行番号だけがちがっていて、文がまったく同じか類似しているものは、カーソルを動かし、行番号だけを修正するか文も修正して⏎キーを押せば、キーインができたことになります。やってみましょう。

インプットデータの修正

経営分析プログラムを実行してデータをインプットする場合、キーを押しまちがえることがあります。⏎キーを押す前であれば、カーソルを動かしてデータを修正することができます。しかし、⏎キーを押

してしまえば、データの修正は簡単にはできません。

そこで、データのインプットをまちがえた場合に、簡単に修正できるようなプログラムにしておくことが大切です。

```
400 PRINT
410 PRINT "data ヲ シュウセイ シマスカ ?"
420 PRINT "                1 --- スル"
430 PRINT "                0 --- シナイ"
440 PRINT "                            ";
450 INPUT CD                    ← INPUT文
460 IF CD=1 GOTO 490            ← IF…GOTO文
470 IF CD=0 GOTO 1000           ← IF…GOTO文
480 GOTO 400                    ← GOTO文
490 PRINT
500 PRINT "ナンバ゛ン ノ data ヲ シュウセイ シマスカ";
510 INPUT I                     ← INPUT文
520 PRINT "(";
530 PRINT USING "##";I;
540 PRINT ") ";A$(I);
550 INPUT B(I)                  ← INPUT文
560 GOTO 310                    ← GOTO文
1000 REM ***** ヒリツ ケイサン *****
```

まず、行番号400～440のＰＲＩＮＴ文で、画面に次のメッセージを表示させます。

```
data ヲ シュウセイ シマスカ ?
          1 --- スル
          0 --- シナイ
                      ?
```

← メッセージ

次は、行番号450のＩＮＰＵＴ文です。データを修正するのであれば1をキーインし、修正しないのであれば0をキーインします。1あるいは0は、変数ＣＤに代入されます。

行番号460、470のＩＦ…ＧＯＴＯ文により、変数ＣＤの値が1（修正）であれば行番号490へ飛び、0（修正しない）であれば行番号1000へ飛びます。なお、ＣＤの値が1あるいは0以外であれば、次の行番号480へ行きます。

そこで、1あるいは0以外の数がキーインされていると、行番号480のＧＯＴＯ文により、行番号400へもどり、再度データを修正するか否かのメッセージが表示されます。

データを修正する（ＣＤ＝１）のであれば、行番号490に飛んできて何番のデータを修正するかをキーインさせ、その値を変数Ｉに代入します。そしてＩ番目の項目名を画面に表示して、正しいデータをキーインさせ、その値を配列変数Ｂ（Ｉ）に代入します。

こうしてデータの修正が終わったら、行番号560のＧＯＴＯ文により、行番号310へもどり、再度インプットデータを画面に表示します。そして、データを修正するか否かのメッセージが画面に表示されるので、さらに修正したいデータがあれば１をキーインすればよいし、すべてのデータが正しくキーインできていれば０をキーインすればよい。

すべてのデータが正しくキーインできていれば、行番号470のＩＦ…ＧＯＴＯ文により、行番号1000へ飛びます。

データの修正部分をまとめておけば、次のとおりです。

```
310 CLS
320 PRINT "*** input data ***"
  (
400 PRINT
410 PRINT "data ｦ ｼｭｳｾｲ ｼﾏｽｶ ?"
420 PRINT "                  1 --- ｽﾙ"
430 PRINT "                  0 --- ｼﾅｲ"
440 PRINT "                              ";
450 INPUT CD
460 IF CD=1 GOTO 490
470 IF CD=0 GOTO 1000
480 GOTO 400
490 PRINT
500 PRINT "ﾅﾝﾊﾞﾝ ﾉ data ｦ ｼｭｳｾｲ ｼﾏｽｶ";
510 INPUT I
520 PRINT "(";
530 PRINT USING "##";I;
540 PRINT ") ";A$(I);
550 INPUT B(I)
560 GOTO 310
1000 REM ***** ﾋﾘﾂ ｹｲｻﾝ *****
```

データを修正する

データを修正しない

5・4 比率計算のプログラム

注釈を入れる

比率計算プログラムは、行番号1000から始めることにします。

ＡＵＴＯコマンドを使って、次のようにキーインすれば行番号が1000から10きざみに、1000、1010、1020、……と自動的に発生します。 **ＡＵＴＯコマンド**

```
(f・2) 1000 ⏎   ← 行番号を 1000，1010，1020，……と自動的に発生させる
```

行番号1 000に、ＲＥＭ文を使って、注釈（コメント）を入れておきます。 **ＲＥＭ文**

```
1000 REM ***** ヒリツ ケイサン *****   ← 注釈（コメント）文
```

比率名を文字型配列変数に代入する

ここで、次の代入文を使って、10の比率名を文字型配列変数に代入します。 **代入文**

```
文字型配列変数＝〝文字列〟  ← ストリング（文字列）を文字型配列変数に代入
```

次のとおりです。

```
1010 C$(1)="ヘイキンソウシホン ケイシ゛ョウリエキリツ"
1020 C$(2)="ウリアケ゛タ゛カ   ケイシ゛ョウリエキリツ"
1030 C$(3)="ヘイキンソウシホン カイテンリツ"
1040 C$(4)="リュウト゛ウヒリツ"
1050 C$(5)="コテイヒリツ"
1060 C$(6)="シ゛コシホンヒリツ"
1070 C$(7)="シ゛ュウキ゛ョウイン ヒトリアタリ ウリアケ゛タ゛カ"
1080 C$(8)="シ゛ュウキ゛ョウイン ヒトリアタリ ケイシ゛ョウリエキ"
1090 C$(9)="ソ゛ウシュウリツ"
1100 C$(10)="ケイシ゛ョウソ゛ウエキリツ"
```
← 比率名を文字型配列変数に代入

これらの代入文の右辺は、すべて、ダブルクォーテーション（″）で囲まれたストリング（文字列）、つまり文字定数ですね。 **文字定数**

また、代入文の左辺の文字型配列変数は、行番号20のＤＩＭ文で配列の宣言がなされていましたね。

これらの代入文を実行すると、右辺のストリング（文字列）が左辺の文字型配列変数に代入されます。

比率計算

そして、次の代入文を使って、10の比率を計算し、その値を配列変数に代入します。

```
配列変数＝式 ⏎   ←―― 式の値を配列変数に代入
```

次のとおりです。

```
1110 D(1)=B(10)/((B(5)+B(6))/2)*100   ←平均総資本経常利益率
1120 D(2)=B(10)/B(8)*100              ←売上高経常利益率
1130 D(3)=B(8)/((B(5)+B(6))/2)        ←平均総資本回転率
1140 D(4)=B(1)/B(3)*100               ←流動比率
1150 D(5)=B(2)/B(4)*100               ←固定比率
1160 D(6)=B(4)/B(6)*100               ←自己資本比率
1170 D(7)=B(8)/((B(11)+B(12))/2)      ←従業員1人あたり売上高
1180 D(8)=B(10)/((B(11)+B(12))/2)     ←従業員1人あたり経常利益
1190 D(9)=(B(8)-B(7))/B(7)*100        ←増収率
1200 D(10)=(B(10)-B(9))/B(9)*100      ←経常増益率
```

10の比率の計算式については、5・1の経営分析プログラムのフローチャートのところを見てください。

代入文の右辺の配列変数B（1）～B（10）の値は、5・3のデータインプットのプログラムを実行すれば、キーインされます。

また、代入文の左辺の配列変数は、行番号20のＤＩＭ文で配列の宣言がなされていましたね。

なお、パソコンでは、カッコは、{ } や〔 〕を使わずに、すべて( )を使います。また、(の数と) の数が同じでなければなりません。

(の数と) の数が同じでないと、プログラムの実行時に、「？ＳＮ Ｅｒｒｏｒ」あるいは「Ｓｙｎｔａｘ　ｅｒｒｏｒ」のエラーメッセージが表示されます。

プログラムをキーインしたら、必ず、(の数と) の数が同じかどうかチェックしてください。

5・5 メニューの選択のプログラム

注釈を入れる

メニューの選択のプログラムは、行番号 2000 から始めることにします。

ＡＵＴＯコマンドを使って、行番号を 2000 から10きざみに自動的に発生させます。　**AUTOコマンド**

```
f・2  2000 ⏎  ←―― 行番号を2000から10きざみに自動的に発生させる
```

行番号 2000 にＲＥＭ文を使って、注釈(コメント) を入れておきます。　**REM文**

```
2000 REM ***** ﾒﾆｭｰ(ﾋﾘﾂ) ﾉ ｾﾝﾀｸ *****  ←注釈(コメント)文
```

画面をクリア

ＣＬＳ文を使って、画面をクリアします。　**CLS文**

```
2010 CLS  ←―― 画面クリア
```

メニューの選択

```
2020 PRINT "ﾄﾞﾉ ﾋﾘﾂ ｦ ｼﾗﾍﾞﾏｽｶ ?"
2030 PRINT
2040 PRINT "                  1 --- ｼｭｳｴｷｾｲ"
2050 PRINT "                  2 --- ｱﾝｾﾞﾝｾｲ"
2060 PRINT "                  3 --- ｾｲｻﾝｾｲ"
2070 PRINT "                  4 --- ｾｲﾁｮｳｾｲ"
2080 PRINT "                  5 --- END"
2090 PRINT "                              ".
2100 INPUT M  ←―― インプット文
```
（2020〜2090：メニュー画面の表示）

まず、行番号 2020～2090 のＰＲＩＮＴ文で、ディスプレイに次のメニュー画面を表示させます。　**PRINT文**

画面は、次ページのようになります。



INPUT文

次は、行番号 2100 のＩＮＰＵＴ文です。

ここで、１をキーインすれば収益性の比率表示、２をキーインすれば安全性の比率表示、３をキーインすれば生産性の比率表示、４をキーインすれば成長性の比率表示をします。５をキーインすれば経営分析プログラムの実行を終了します。

キーインした数値は、変数Ｍに代入されます。

画面のクリア

CLS文

ここで、ＣＬＳ文を使って、画面をクリアします。

```
2110 CLS          ← 画面クリア
```

ジャンプ先は？

```
2120 IF M=1 GOTO 2180   ← M＝1なら、行番号2180へ飛べ
2130 IF M=2 GOTO 2260   ← M＝2なら、行番号2260へ飛べ。
2140 IF M=3 GOTO 2340   ← M＝3なら、行番号2340へ飛べ
2150 IF M=4 GOTO 2400   ← M＝4なら、行番号2400へ飛べ
2160 IF M=5 GOTO 2460   ← M＝5なら、行番号2460へ飛べ
2170 GOTO 2000          ← Mがその他の値なら、行番号2000へ飛べ
```

Ｍ＝１（収益性）なら行番号 2180 へ飛び、Ｍ＝２（安全性）なら行番号 2260 へ飛び、Ｍ＝３（生産性）なら行番号 2340 へ飛び、Ｍ＝４（成長性）なら行番号 2400 へ飛び、Ｍ＝５（ＥＮＤ）なら行番号 2460 へ飛びます。

しかし、Ｍの値がその他の値なら、行番号 2000 へ飛び、メニュー画面の表示へもどります。

比率の表示

```
2180 PRINT "*** ｼｭｳｴｷｾｲ ***":PRINT
2190 PRINT :PRINT C$(1):PRINT
2200 PRINT USING "            ###.##% ";D(1)
2210 PRINT :PRINT C$(2):PRINT
2220 PRINT USING "            ###.##% ";D(2)
2230 PRINT :PRINT C$(3):PRINT
2240 PRINT USING "            ###.##ｶｲ";D(3)
2250 GOTO 2490
2260 PRINT "*** ｱﾝｾﾞﾝｾｲ ***":PRINT
2270 PRINT :PRINT C$(4):PRINT
2280 PRINT USING "            ###.##% ";D(4)
2290 PRINT :PRINT C$(5):PRINT
2300 PRINT USING "            ###.##% ";D(5)
2310 PRINT :PRINT C$(6):PRINT
2320 PRINT USING "            ###.##% ";D(6)
2330 GOTO 2490
2340 PRINT "*** ｾｲｻﾝｾｲ ***":PRINT
2350 PRINT :PRINT C$(7):PRINT
2360 PRINT USING "            ###.##  ";D(7)
2370 PRINT :PRINT C$(8):PRINT
2380 PRINT USING "            ###.##  ";D(8)
2390 GOTO 2490
2400 PRINT "*** ｾｲﾁｮｳｾｲ ***":PRINT
2410 PRINT :PRINT C$(9):PRINT
2420 PRINT USING "            ###.##% ";D(9)
2430 PRINT :PRINT C$(10):PRINT
2440 PRINT USING "            ###.##% ";D(10)
2450 GOTO 2490
```

収益性の比率表示（2180〜2250）
安全性の比率表示（2260〜2330）
生産性の比率表示（2340〜2390）
成長性の比率表示（2400〜2450）

「ずいぶん長いな」と思いませんか。しかし、よく見れば、収益性の比率表示と他の比率表示はほとんど同じですね。ちがうところは、行番号、見出し、文字型配列変数と配列変数の添字、%とカイなどですね。キーインのくふうをすれば、以外に簡単にキーインできます。

ＰＲＩＮＴ文は、マルチステートメントになっていますね。３つのＰＲＩＮＴ文が１つのマルチステートメントになっているものもありますね。ＰＲＩＮＴ　ＵＳＩＮＧ文もありますね。

比率表示がすめば、ＧＯＴＯ文により、行番号2490へ飛びます。

ＰＲＩＮＴ文

ＰＲＩＮＴ　ＵＳＩＮＧ文

ＧＯＴＯ文

ゴクロウサマデシタ

```
2160 IF M=5 GOTO 2460
  ~
2460 PRINT "ゴクロウサマデシタ"
2470 PRINT
2480 PRINT "ケイエイブンセキ ヲ オワリマス"
```

行番号2160のＩＦ…ＧＯＴＯ文により、Ｍ＝５（ＥＮＤ）なら行番号2460に飛んできます。

ＰＲＩＮＴ文

行番号2460～2480のＰＲＩＮＴ文で、画面に次のメッセージを表示させます。

```
ゴクロウサマデシタ
ケイエイブンセキ ヲ オワリマス
```

← メッセージ

ＨＩＴ ＡＮＹＫＥＹ

```
2250 GOTO 2490
  ~
2330 GOTO 2490
  ~
2390 GOTO 2490
  ~
2450 GOTO 2490
  ~
2480 PRINT "ケイエイブンセキ ヲ オワリマス"
2490 LOCATE 10,17
2500 PRINT "--- HIT ANYKEY ---"
2510 Z$=INKEY$
2520 IF Z$="" GOTO 2490
2530 BEEP
2540 IF M=5 GOTO 3000
2550 GOTO 2000
```

LOCATE文

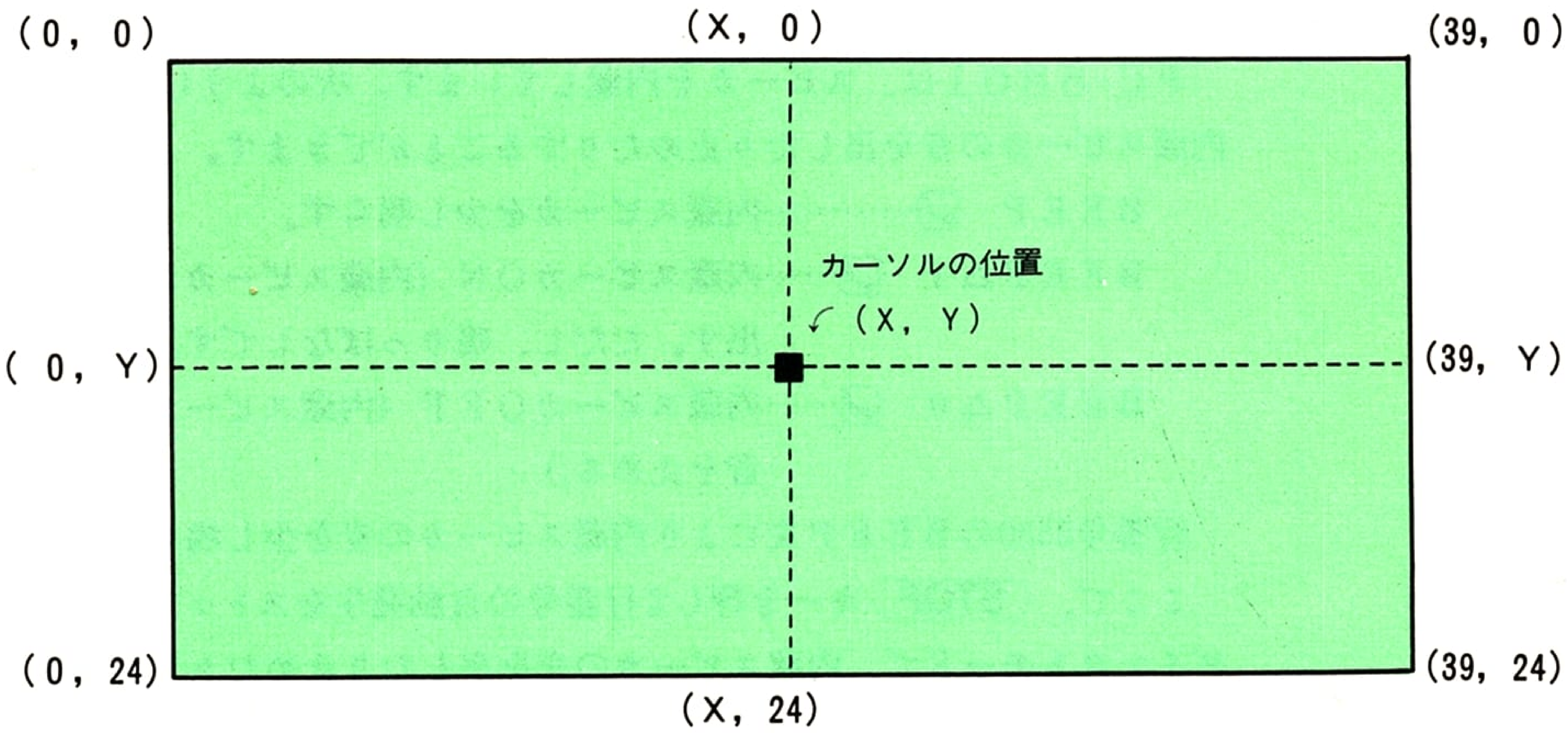
行番号2490の文（ステートメント）は、ＬＯＣＡＴＥ文です。

行番号2250、2330、2390、2450のＧＯＴＯ文を実行した場合、あるいは行番号2480のＰＲＩＮＴ文を実行すると、このＬＯＣＡＴＥ文にきます。

ＬＯＣＡＴＥ文は、画面のカーソルの位置を「Ｘ桁めでＹ行め」に指定するもので、次の形をしています。

LOCATE △ X,Y ⏎	← 画面のカーソルの位置をＸ桁めでＹ行めに指定

行番号40のＷＩＤＴＨコマンドで、画面を40桁25行に指定していました。40桁25行の画面の座標は、次のようになります。なお、画面の左上が「０桁めで０行め（０，０）」です。



行番号2490のＬＯＣＡＴＥ文によって、画面のカーソルの位置を10桁めで17行めに指定します。

PRINT文

次に、行番号2500のＰＲＩＮＴ文で、画面の10桁め17行めから、「−−− ＨＩＴ ＡＮＹＫＥＹ −−−」とメッセージを表示させます。

代入文

行番号2510の代入文には、ＩＮＫＥＹ＄関数が使われています。

INKEY$関数

ヌルストリング

ＩＮＫＥＹ＄関数は、キーが押されていれば文字型変数にその文字を与えます。キーが押されなければ、ヌルストリング（空の文字列）を文字型変数に与えます。

ＩＮＫＥＹ＄関数は、次の形をしています。

文字型変数＝INKEY$ ⏎	← キーが押されていれば文字型変数にその文字を与え、キーが押されなければ文字型変数にヌルストリングを与える

したがって、行番号2510の代入文は、キーが押されていればその文字を文字型変数Ｚ＄に代入します。キーが押されなければ、Ｚ＄にヌルストリングを代入します。

ＩＦ…ＧＯＴＯ文
ヌルストリング

行番号2520は、ＩＦ…ＧＯＴＯ文です。ここで、Ｚ＄がヌルストリングであれば、行番号2490へもどります。なお、ダブルクォーテーション（"）が２つ並んでいる文字列のない文字列は、ヌルストリング（空の文字列）です。

ループ

すなわち、行番号2490～2520はループになっています。まず、画面に「―――　ＨＩＴ　ＡＮＹＫＥＹ　―――」のメッセージが表示されます。ここで、キーを押さなければ、画面の同じ位置に同じメッセージが表示されます。キーを押さないでいると、このループが何回も繰り返し実行され、見た目には、画面に「―――　ＨＩＴ　ＡＮＹＫＥＹ　―――」のメッセージが表示されたままになっています。どれかキーを押すと、このループからぬけ出し、次の行番号2530へ行きます。

ＢＥＥＰ文

行番号2530は、ＢＥＥＰ文です。

ＰＣ-８８０１は、スピーカを内蔵しています。次のようにすれば、内蔵スピーカの音を出したり止めたりすることができます。

ＢＥＥＰ ⏎…………内蔵スピーカを少し鳴らす。
ＢＥＥＰ△１ ⏎……内蔵スピーカＯＮ（内蔵スピーカから音を出す。ただし、鳴りっぱなしです。）
ＢＥＥＰ△０ ⏎……内蔵スピーカＯＦＦ（内蔵スピーカからの音を止める。）

行番号2530のＢＥＥＰ文により内蔵スピーカの音を少し鳴らします。

ここで、（STOP）キーを押して行番号の自動発生をストップさせ、ダイレクトモードで、内蔵スピーカの音を出したり止めたりしてみましょう。まず、次のようにキーインします。

```
BEEP △ ⏎  ←―――――― 内蔵スピーカの音を少し鳴らす
```

内蔵スピーカの音が少し鳴りますね。

次に、ＢＥＥＰ△１⏎とキーインしてみましょう。

```
BEEP △1 ⏎  ←―――――― 内蔵スピーカ鳴りっぱなし！
```

内蔵スピーカは鳴りっぱなしですね。

次のようにキーインすれば、止まります。

```
BEEP △0 ⏎  ←―――――― 内蔵スピーカを止める。
```

ＢＥＥＰ文をうまく使えば、プログラムを実行して楽しいものになります。プログラムを作成するとき、くふうしてみましょう。

ＩＦ…ＧＯＴＯ文

さて、行番号2540は、ＩＦ…ＧＯＴＯ文です。変数Ｍの値が５（ＥＮＤ）であれば、行番号3000へ飛びます。Ｍ＝５でなければ、次の行番号2550へ行き、ＧＯＴＯ文により行番号2000へもどります。

ＧＯＴＯ文

5·6 エンド部分のプログラム

画面のクリア

いよいよ、経営分析プログラムも終わりです。エンド（終わり）部分のプログラムは、行番号3000から始めることにします。ＡＵＴＯコマンドを使って、行番号3000から10きざみに自動的に発生させます。

ＡＵＴＯコマンド

```
(f·2) 3000 ⏎  ←—— 行番号を3000から10きざみに自動的に発生させる
```

ＣＬＳ文を使って、画面をクリアします。

ＣＬＳ文

```
2540 IF M=5 GOTO 3000
  ∫
3000 CLS
```

行番号2540のＩＦ…ＧＯＴＯ文により、Ｍ＝５（ＥＮＤ）なら行番号3000のＣＬＳ文に飛んできます。

画面の桁数、行数の指定

ＷＩＤＴＨコマンドを使って、画面の桁数、行数を指定します。ここでは、次のようにキーインしておくことにします。

ＷＩＤＴＨコマンド

```
3010 WIDTH 80,20  ←—— 画面を80桁20行にする
```

画面のモード設定を行う

次に、ＣＯＮＳＯＬＥ文を使って、画面モード設定を行います。ここでは、次のようにキーインしておきます。

ＣＯＮＳＯＬＥ文

```
3020 CONSOLE 0,20,1,0  ←—— ファンクションキーの表示をする
```

経営分析プログラムの実行を終了させる

最後に、ＥＮＤ文を使って、経営分析プログラムの実行を終了させます。

ＥＮＤ文

```
3030 END  ←—— 経営分析プログラムの実行を終了させる
```

5・7 経営分析プログラムのリスト

経営分析プログラムのリストを画面に表示する

LISTコマンド

ＬＩＳＴコマンドを使って、経営分析プログラムのリストを画面に表示してみましょう。 f･4 キーを押し、⏎キーを押せばよかったですね。

f･4 ⏎ ←	経営分析プログラムのリストを画面に表示

経営分析プログラムのリストが、画面の下から上へと流れていきます。この流れを一時ストップさせるには、 CTRL キーを押しながらSキーを押せばよかったですね。

CTRL ＋ S ←	画面の表示を一時ストップさせる

画面の表示が一時ストップしますね。表示を再開させるには、どれでもいいですから、どれかキーを押せばよいのです。やってみましょう。

経営分析プログラムのリストをプリンタに出力する

LLISTコマンド

ＬＬＩＳＴコマンドを使って、経営分析プログラムのリストをプリンタに出力してみましょう。次のようにキーインしましょう。

1 f･4 ⏎ ←	経営分析プログラムのリストをプリンタに出力

経営分析プログラムのリストは、次のとおりです。

ずいぶん長いプログラムですね。しかし、大部分がＰＲＩＮＴ文だと思いませんか。見やすい画面にするために苦労しているわけです。

```
10 REM ***** ｹｲｴｲ ﾌﾞﾝｾｷ *****            ← スタート部分
20 DIM A$(12),B(12),C$(10),D(10)
30 CLS
40 WIDTH 40,25
50 CONSOLE 0,25,0,0
60 PRINT "********* ｹｲｴｲ ﾌﾞﾝｾｷ *********"
70 PRINT
100 REM ***** DATA ﾉ INPUT *****        ← データインプット
110 A$(1)="ﾘｭｳﾄﾞｳｼｻﾝ                    :
120 A$(2)="ｺﾃｲｼｻﾝ                       :
130 A$(3)="ﾘｭｳﾄﾞｳﾌｻｲ                    :
```

```
140 A$(4)="ｼﾞｺｼﾎﾝ                    "
150 A$(5)="ｷｼｭ ｿｳｼﾎﾝ                 "
160 A$(6)="ｷﾏﾂ ｿｳｼﾎﾝ                 "
170 A$(7)="ｾﾞﾝｷ ｳﾘｱｹﾞﾀﾞｶ             "
180 A$(8)="ﾄｳｷ  ｳﾘｱｹﾞﾀﾞｶ             "
190 A$(9)="ｾﾞﾝｷ ｹｲｼﾞｮｳﾘｴｷ            "
200 A$(10)="ﾄｳｷ  ｹｲｼﾞｮｳﾘｴｷ            "
210 A$(11)="ｷｼｭ ｼﾞｭｳｷﾞｮｳｲﾝｽｳ          "
220 A$(12)="ｷﾏﾂ ｼﾞｭｳｷﾞｮｳｲﾝｽｳ          "
230 PRINT "--- data ｦ input ｼﾃｸﾀﾞｻｲ ---"
240 PRINT
250 FOR I=1 TO 12
260 PRINT "(";
270 PRINT USING "##";I;
280 PRINT ") ";A$(I);
290 INPUT B(I)
300 NEXT I
310 CLS
320 PRINT "*** input data ***"
330 PRINT
340 FOR I=1 TO 12
350 PRINT "(";
360 PRINT USING "##";I;
370 PRINT ") ";A$(I);
380 PRINT USING "#,###,###";B(I)
390 NEXT I
400 PRINT
410 PRINT "data ｦ ｼｭｳｾｲ ｼﾏｽｶ ?"
420 PRINT "                 1 --- ｽﾙ"
430 PRINT "                 0 --- ｼﾅｲ"
440 PRINT "                                  ";
450 INPUT CD
460 IF CD=1 GOTO 490
470 IF CD=0 GOTO 1000
480 GOTO 400
490 PRINT
500 PRINT "ﾅﾝﾊﾞﾝ ﾉ data ｦ ｼｭｳｾｲ ｼﾏｽｶ";
510 INPUT I
520 PRINT "(";
530 PRINT USING "##";I;
540 PRINT ") ";A$(I);
550 INPUT B(I)
560 GOTO 310
```

```
1000 REM ***** ﾋﾘﾂ ｹｲｻﾝ *****  ←比率計算
1010 C$(1)="ﾍｲｷﾝｿｳｼﾎﾝ ｹｲｼﾞｮｳﾘｴｷﾘﾂ           "
1020 C$(2)="ｳﾘｱｹﾞﾀﾞｶ   ｹｲｼﾞｮｳﾘｴｷﾘﾂ           "
1030 C$(3)="ﾍｲｷﾝｿｳｼﾎﾝ ｶｲﾃﾝﾘﾂ                 "
1040 C$(4)="ﾘｭｳﾄﾞｳﾋﾘﾂ                        "
1050 C$(5)="ｺﾃｲﾋﾘﾂ                           "
1060 C$(6)="ｼﾞｺｼﾎﾝﾋﾘﾂ                        "
1070 C$(7)="ｼﾞｭｳｷﾞｮｳｲﾝ ﾋﾄﾘｱﾀﾘ ｳﾘｱｹﾞﾀﾞｶ "
1080 C$(8)="ｼﾞｭｳｷﾞｮｳｲﾝ ﾋﾄﾘｱﾀﾘ ｹｲｼﾞｮｳﾘｴｷ"
1090 C$(9)="ｿﾞｳｼｭｳﾘﾂ                         "
1100 C$(10)="ｹｲｼﾞｮｳｿﾞｳｴｷﾘﾂ                     "
1110 D(1)=B(10)/((B(5)+B(6))/2)*100
1120 D(2)=B(10)/B(8)*100
1130 D(3)=B(8)/((B(5)+B(6))/2)
1140 D(4)=B(1)/B(3)*100
1150 D(5)=B(2)/B(4)*100
1160 D(6)=B(4)/B(6)*100
1170 D(7)=B(8)/((B(11)+B(12))/2)
1180 D(8)=B(10)/((B(11)+B(12))/2)
1190 D(9)=(B(8)-B(7))/B(7)*100
1200 D(10)=(B(10)-B(9))/B(9)*100
2000 REM ***** ﾒﾆｭｰ(ﾋﾘﾂ) ﾉ ｾﾝﾀｸ *****  ←メニューの選択
2010 CLS
2020 PRINT "ﾄﾞﾉ ﾋﾘﾂ ｦ ｼﾗﾍﾞﾏｽｶ ?"
2030 PRINT
2040 PRINT "                1 --- ｼｭｳｴｷｾｲ"
2050 PRINT "                2 --- ｱﾝｾﾞﾝｾｲ"
2060 PRINT "                3 --- ｾｲｻﾝｾｲ"
2070 PRINT "                4 --- ｾｲﾁｮｳｾｲ"
2080 PRINT "                5 --- END"
2090 PRINT "                                ";
2100 INPUT M
2110 CLS
2120 IF M=1 GOTO 2180
2130 IF M=2 GOTO 2260
2140 IF M=3 GOTO 2340
2150 IF M=4 GOTO 2400
2160 IF M=5 GOTO 2460
2170 GOTO 2000
2180 PRINT "*** ｼｭｳｴｷｾｲ ***":PRINT
2190 PRINT :PRINT C$(1):PRINT
2200 PRINT USING "               ###.##% ";D(1)
2210 PRINT :PRINT C$(2):PRINT
```

```
2220 PRINT USING "                    ###.##% ";D(2)
2230 PRINT :PRINT C$(3):PRINT
2240 PRINT USING "                    ###.##ｶｲ";D(3)
2250 GOTO 2490
2260 PRINT "*** ｱﾝｾﾞﾝｾｲ ***":PRINT
2270 PRINT :PRINT C$(4):PRINT
2280 PRINT USING "                    ###.##% ";D(4)
2290 PRINT :PRINT C$(5):PRINT
2300 PRINT USING "                    ###.##% ";D(5)
2310 PRINT :PRINT C$(6):PRINT
2320 PRINT USING "                    ###.##% ";D(6)
2330 GOTO 2490
2340 PRINT "*** ｾｲｻﾝｾｲ ***":PRINT
2350 PRINT :PRINT C$(7):PRINT
2360 PRINT USING "                    ###.##  ";D(7)
2370 PRINT :PRINT C$(8):PRINT
2380 PRINT USING "                    ###.##  ";D(8)
2390 GOTO 2490
2400 PRINT "*** ｾｲﾁｮｳｾｲ ***":PRINT
2410 PRINT :PRINT C$(9):PRINT
2420 PRINT USING "                    ###.##% ";D(9)
2430 PRINT :PRINT C$(10):PRINT
2440 PRINT USING "                    ###.##% ";D(10)
2450 GOTO 2490
2460 PRINT "ｺﾞｸﾛｳｻﾏﾃﾞｼﾀ"
2470 PRINT
2480 PRINT "ｹｲｴｲﾌﾞﾝｾｷ ｦ ｵﾜﾘﾏｽ"
2490 LOCATE 10,17
2500 PRINT "--- HIT ANYKEY ---"
2510 Z$=INKEY$
2520 IF Z$="" GOTO 2490
2530 BEEP
2540 IF M=5 GOTO 3000
2550 GOTO 2000
3000 CLS                    ← エンド部分
3010 WIDTH 80,20
3020 CONSOLE 0,20,1,0
3030 END
```

5·8 経営分析プログラムの実行

RUNコマンド

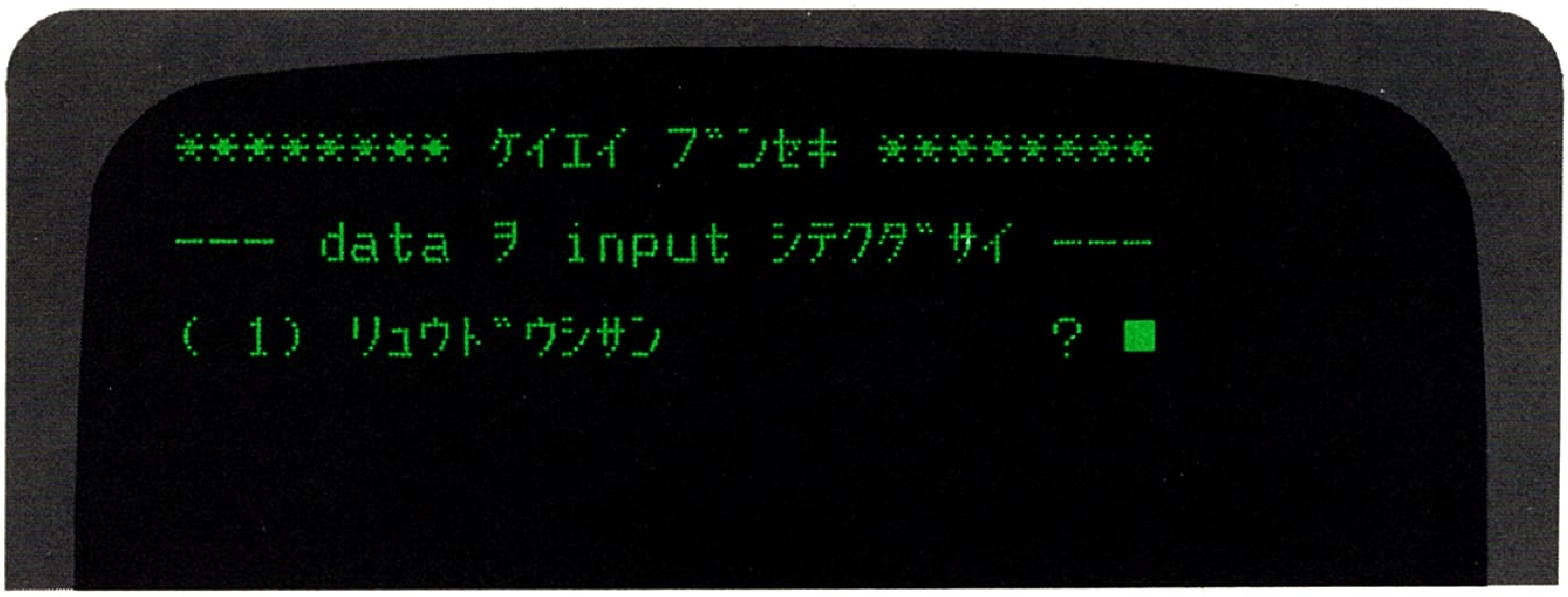
RUN

RUNコマンドを使って、経営分析プログラムを実行してみましょう。f·5 キーを押します。

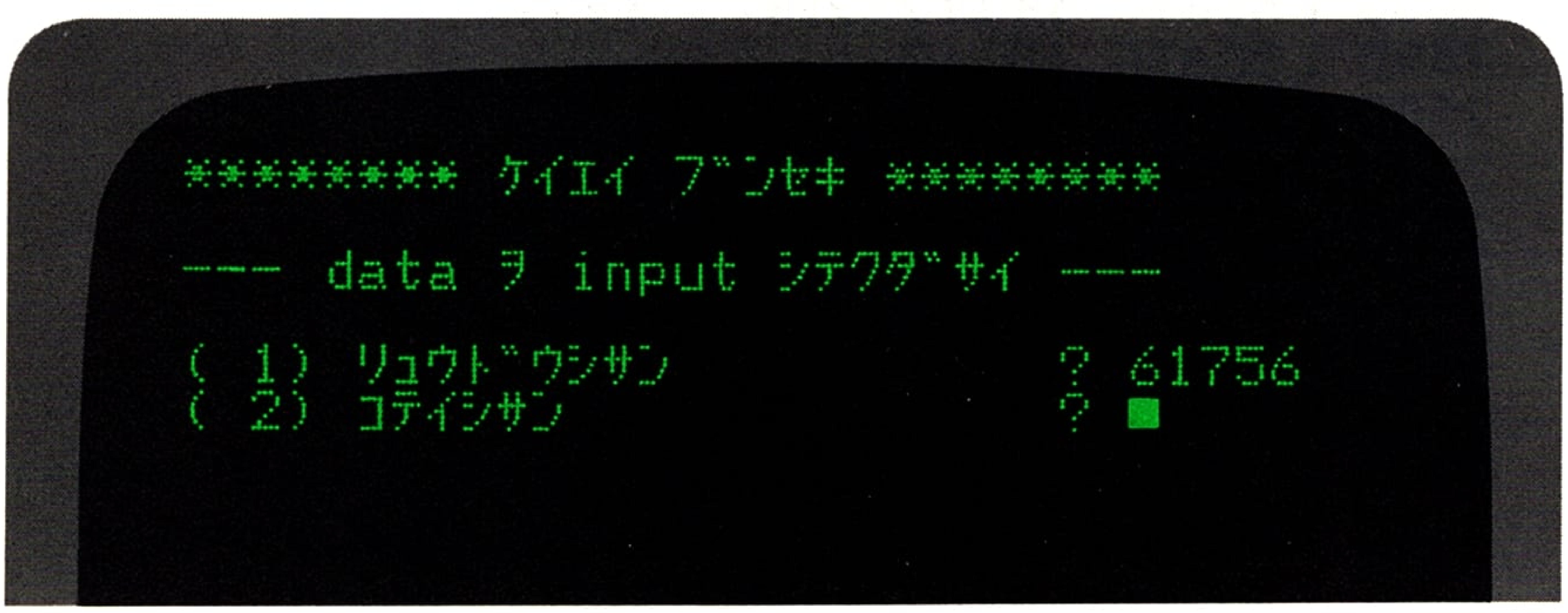
f·5 ←―――― 経営分析プログラムの実行

データのインプット

経営分析プログラムを実行すると、ディスプレイの画面には、次のようにメッセージが表示されます。



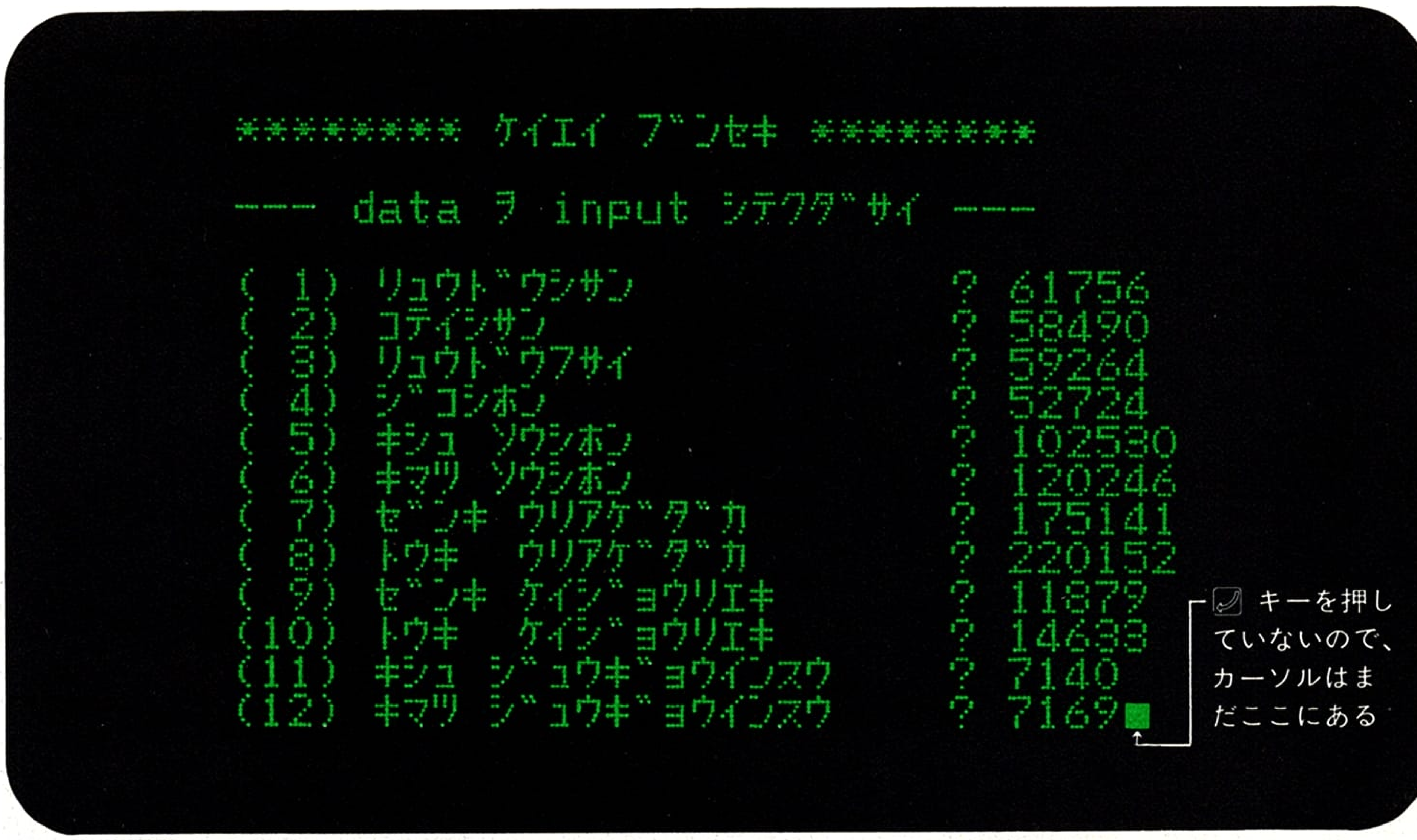
ここで、流動資産のデータ61756(百万円) をキーインし、⏎キーを押すと、画面は次のようになります。



以下、固定資産、流動負債、……、期末従業員数のデータをキーインします。

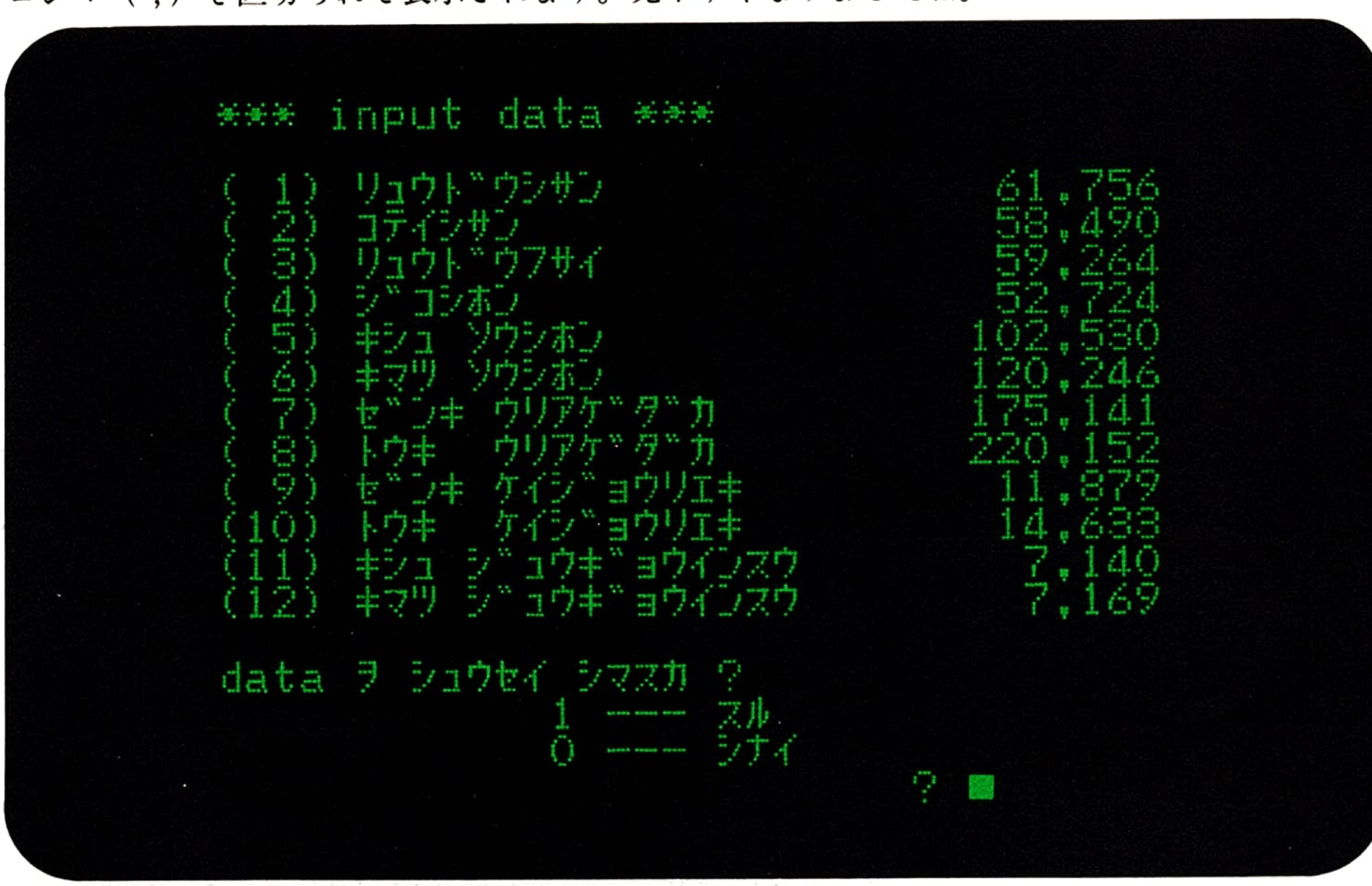
キーインをまちがった場合は、⏎キーを押す前であればカーソルを動かしてデータを修正することができます。しかし、⏎キーを押してしまった場合は、まちがった個所はそのままにしておいて、データのキーインを続行します。

いま、次の画面のようにデータをキーインしたとしましょう。ただし、⑿の期末従業員数のデータ 7169（人）をキーインしたあと、まだ⏎キーを押していません。



インプットデータの見やすい画面表示

ここで、⏎キーを押すと、インプットデータが右づめで 3 桁ごとにコンマ（,）で区切られて表示されます。見やすくなりましたね。



インプットデータの修正

インプットデータを修正するのであれば、[1]キーを押し、⏎キーを押します。次の画面になります。

```
*** input data ***

( 1) ﾘｭｳﾄﾞｳｼｻﾝ                    61,756
( 2) ｺﾃｲｼｻﾝ                       58,490
( 3) ﾘｭｳﾄﾞｳﾌｻｲ                    59,264
( 4) ｼﾞｺｼﾎﾝ                       52,724
( 5) ｷｼｭ ｿｳｼﾎﾝ                   102,530
( 6) ｷﾏﾂ ｿｳｼﾎﾝ                   120,246
( 7) ｾﾞﾝｷ ｳﾘｱｹﾞﾀﾞｶ               175,141
( 8) ﾄｳｷ  ｳﾘｱｹﾞﾀﾞｶ               220,152
( 9) ｾﾞﾝｷ ｹｲｼﾞｮｳﾘｴｷ               11,879
(10) ﾄｳｷ  ｹｲｼﾞｮｳﾘｴｷ               14,688
(11) ｷｼｭ ｼﾞｭｳｷﾞｮｳｲﾝｽｳ              7,140
(12) ｷﾏﾂ ｼﾞｭｳｷﾞｮｳｲﾝｽｳ              7,169

data ｦ ｼｭｳｾｲ ｼﾏｽｶ ?
          1 --- ｽﾙ
          0 --- ｼﾅｲ
                         ? 1

ﾅﾝﾊﾞﾝ ﾉ data ｦ ｼｭｳｾｲ ｼﾏｽｶ? ■
```

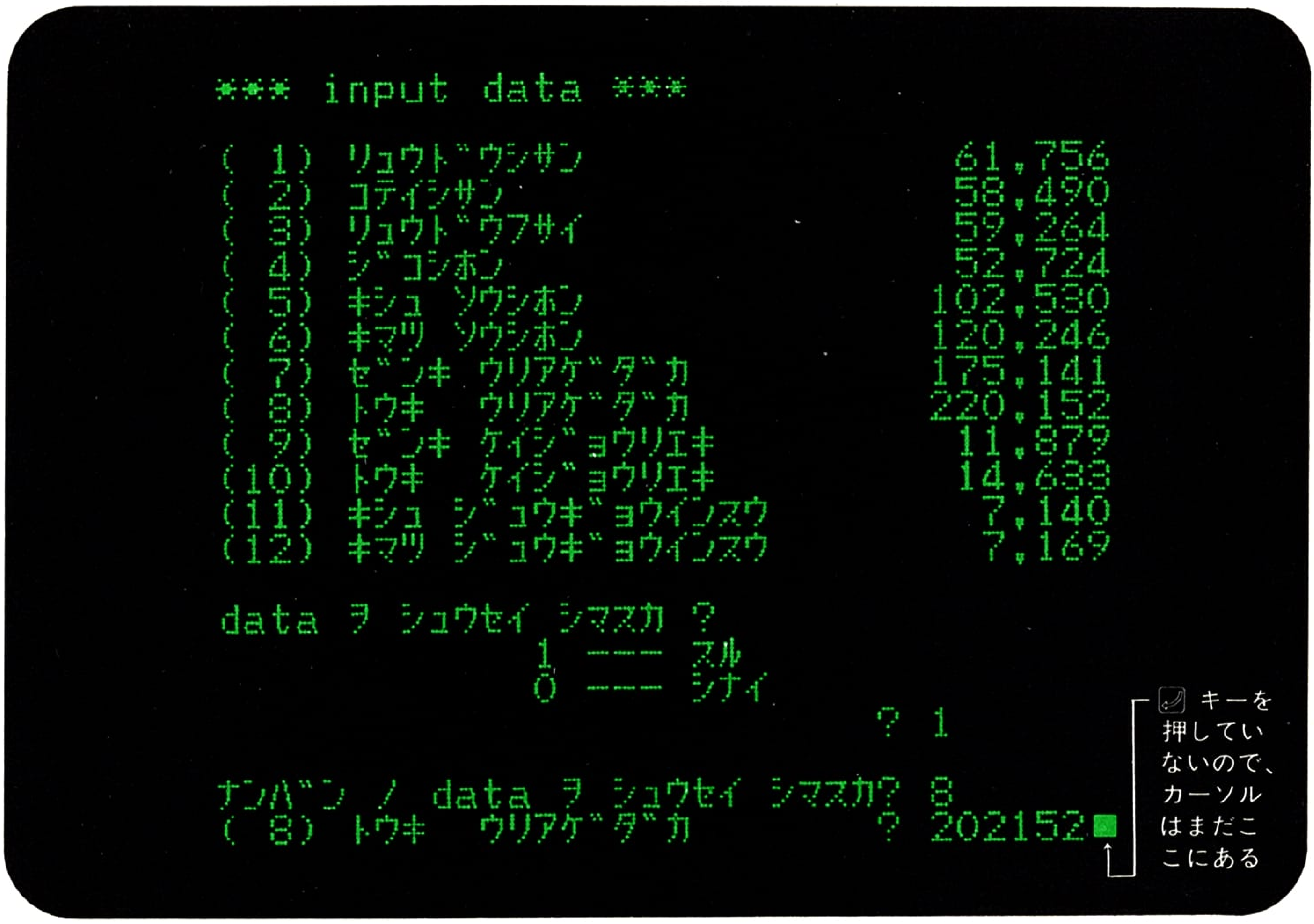
(8)の当期売上高のデータを修正することにしましょう。[8]キーを押し、⏎キーを押します。次の画面になります。

```
( 1) ﾘｭｳﾄﾞｳｼｻﾝ                    61,756
( 2) ｺﾃｲｼｻﾝ                       58,490
( 3) ﾘｭｳﾄﾞｳﾌｻｲ                    59,264
( 4) ｼﾞｺｼﾎﾝ                       52,724
( 5) ｷｼｭ ｿｳｼﾎﾝ                   102,530
( 6) ｷﾏﾂ ｿｳｼﾎﾝ                   120,246
( 7) ｾﾞﾝｷ ｳﾘｱｹﾞﾀﾞｶ               175,141
( 8) ﾄｳｷ  ｳﾘｱｹﾞﾀﾞｶ               220,152
( 9) ｾﾞﾝｷ ｹｲｼﾞｮｳﾘｴｷ               11,879
(10) ﾄｳｷ  ｹｲｼﾞｮｳﾘｴｷ               14,688
(11) ｷｼｭ ｼﾞｭｳｷﾞｮｳｲﾝｽｳ              7,140
(12) ｷﾏﾂ ｼﾞｭｳｷﾞｮｳｲﾝｽｳ              7,169

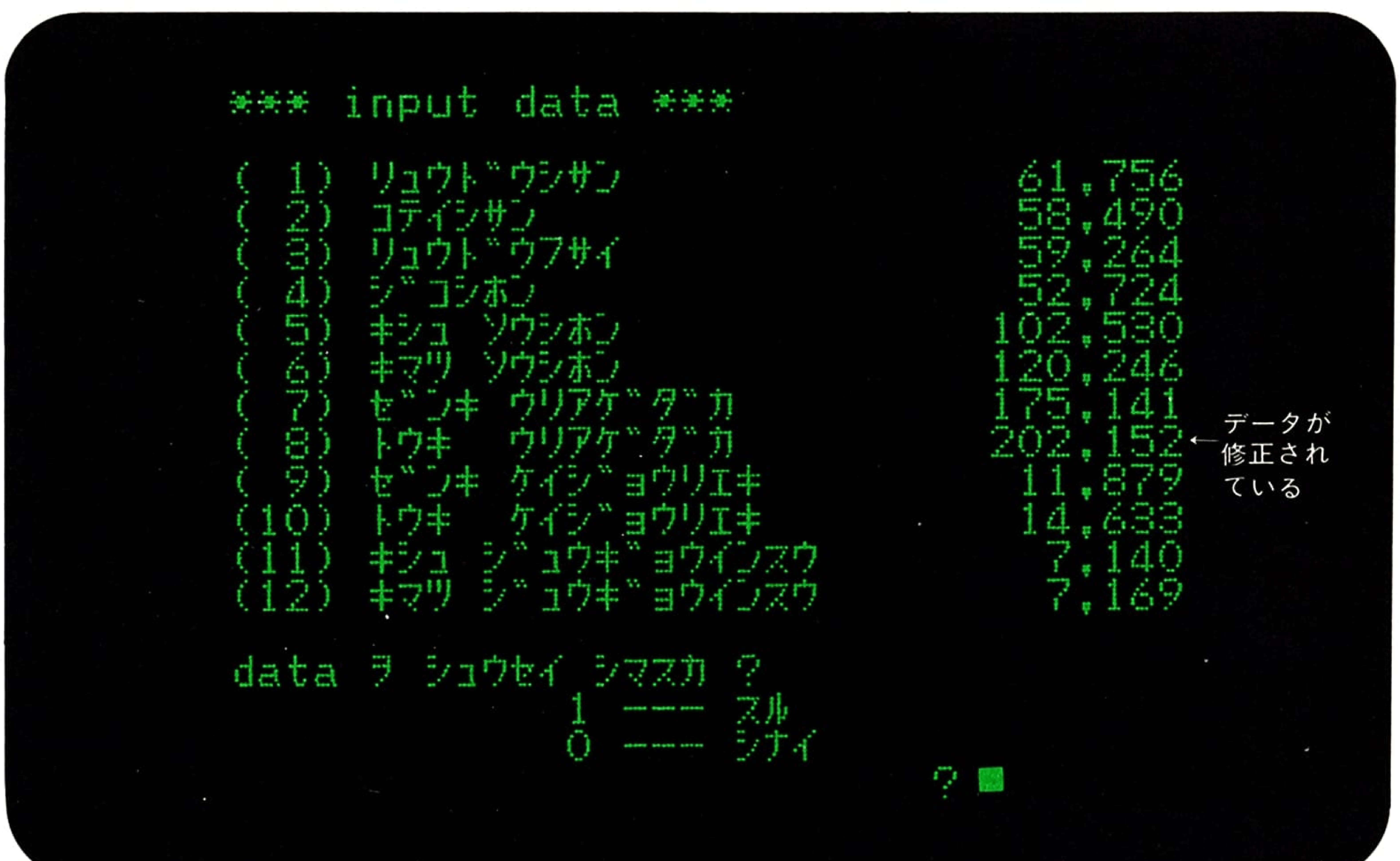
data ｦ ｼｭｳｾｲ ｼﾏｽｶ ?
          1 --- ｽﾙ
          0 --- ｼﾅｲ
                         ? 1

ﾅﾝﾊﾞﾝ ﾉ data ｦ ｼｭｳｾｲ ｼﾏｽｶ? 8
( 8) ﾄｳｷ  ｳﾘｱｹﾞﾀﾞｶ         ? ■
```

(8)当期売上高のデータとして、202152（百万円）をキーインします。ただし、まだ⏎キーを押していません。

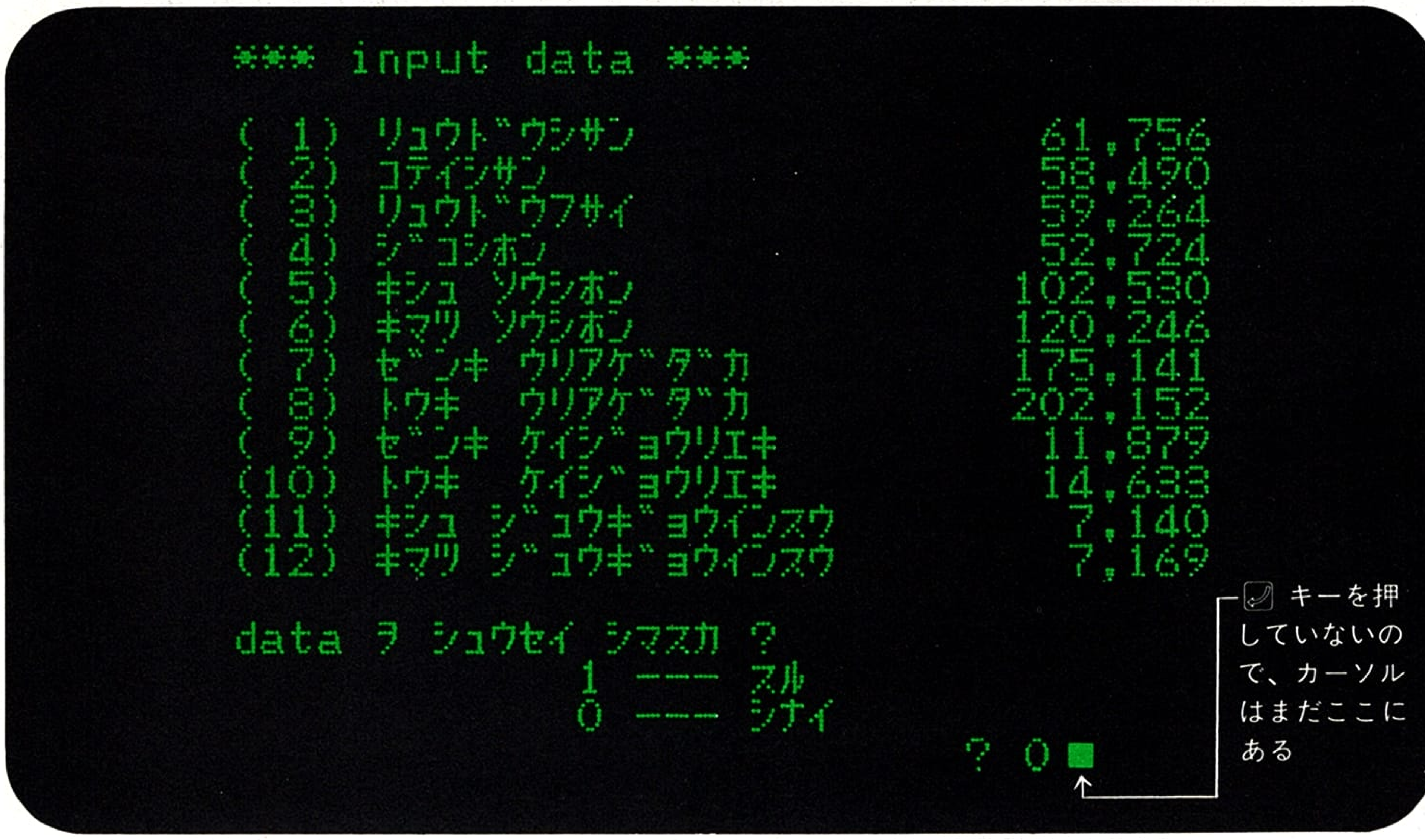


ここで、⏎キーを押せば、次の画面になります。(8)の当期売上高のデータが修正されていますね。



さらに修正したいデータがあれば、1 ⏎とし、修正したいデータの番号をキーインして、正しいデータをキーインすればよいのです。

すべてのデータが正しくキーインできたら、0キーを押します。ただし、まだ⏎キーを押していません。



ここで、⏎キーを押すと、次のメニュー画面が表示されます。

メニュー画面の表示

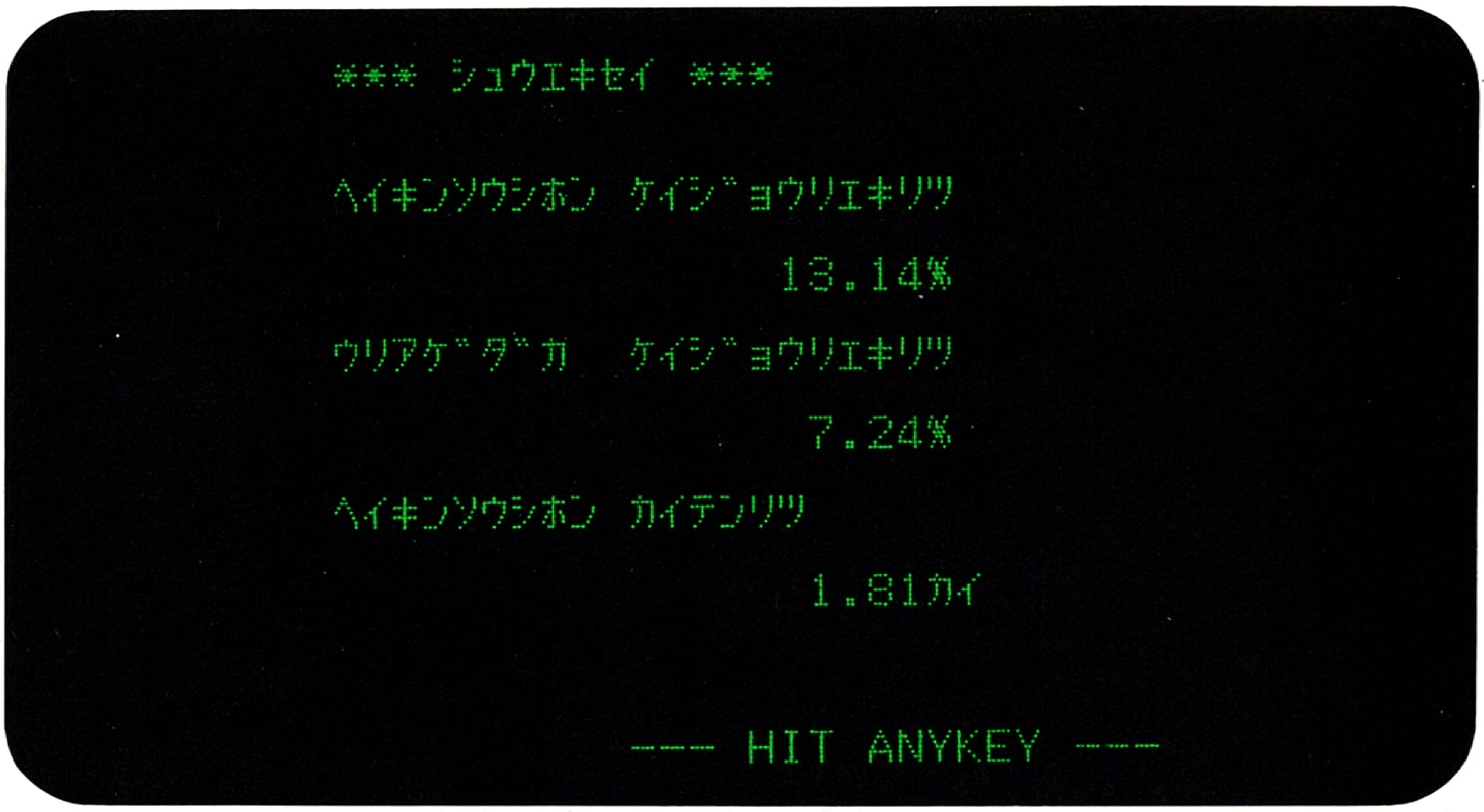
メニュー画面は次のとおりです。



収益性の比率表示

収益性の比率表示

メニュー画面において、[1] [⏎]とキーインすれば、収益性の比率表示をすることができます。

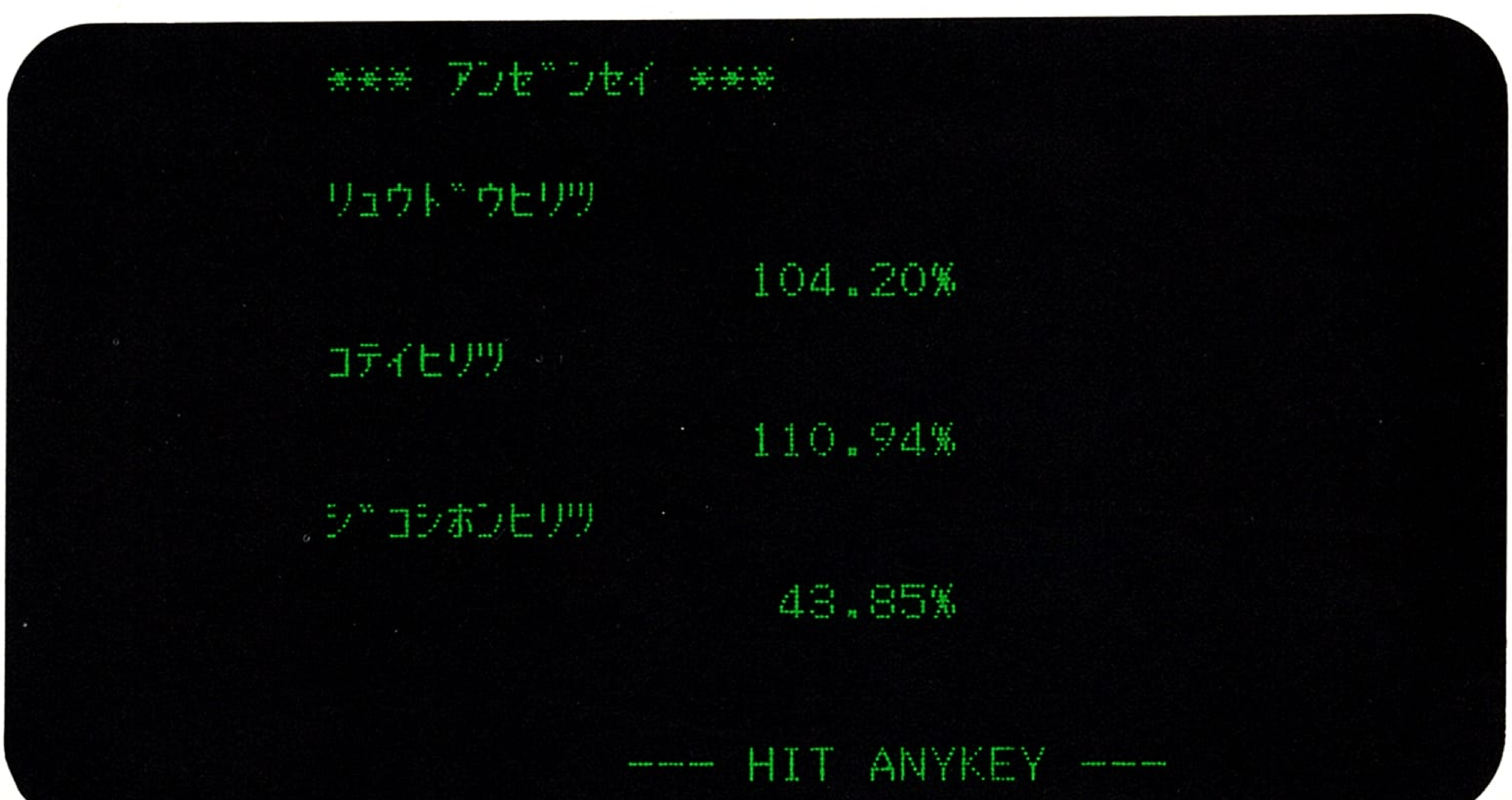


ここで、どのキーでもいいから押せば、メニュー画面にもどります。

安全性の比率表示

安全性の比率表示

メニュー画面において、[2] [⏎]とキーインすれば、安全性の比率が表示されます。

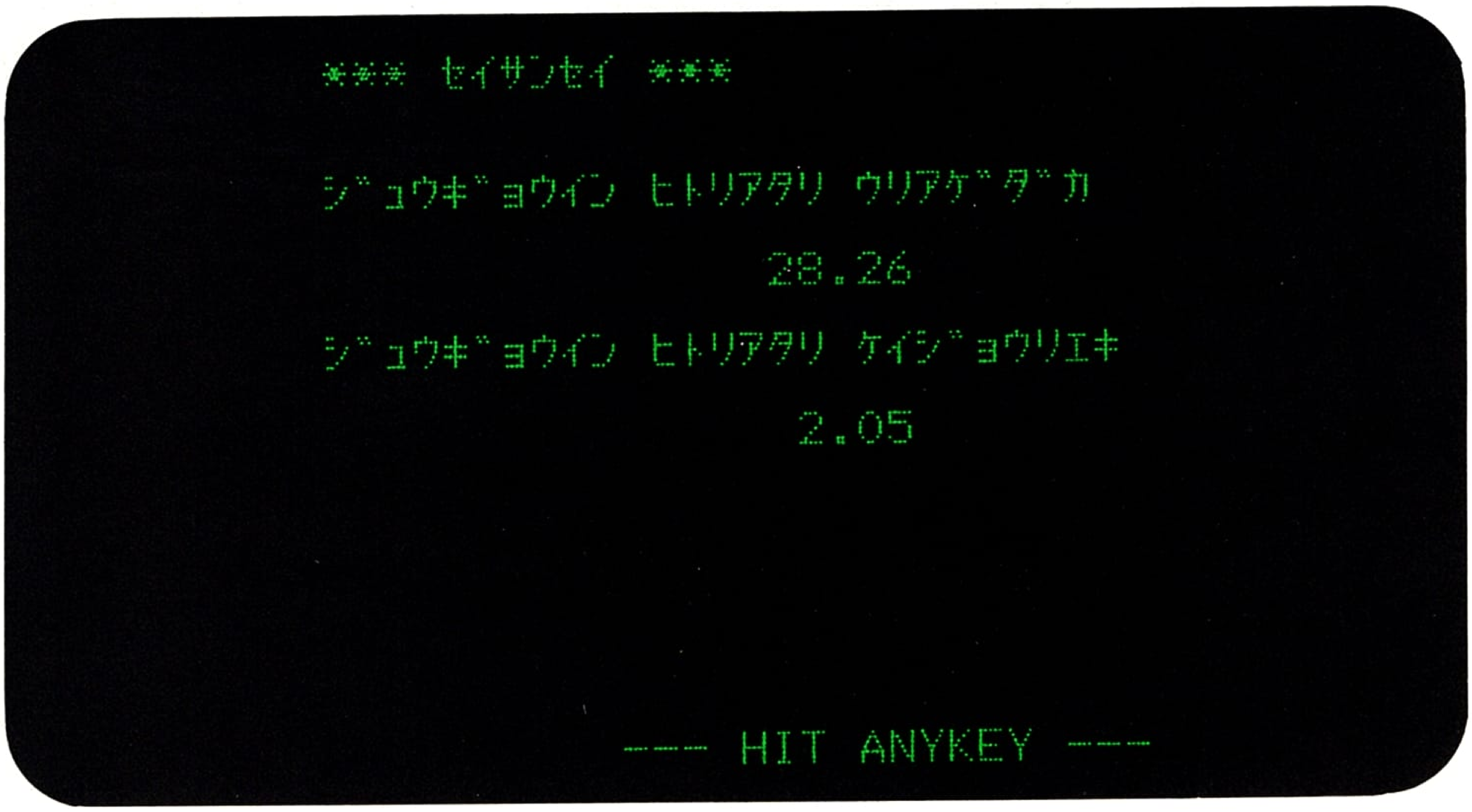


どれかキーを押しましょう。メニュー画面にもどります。

生産性の比率表示

生産性の比率表示

メニュー画面において、3 ⏎とキーインすれば、生産性の比率表示をすることができます。

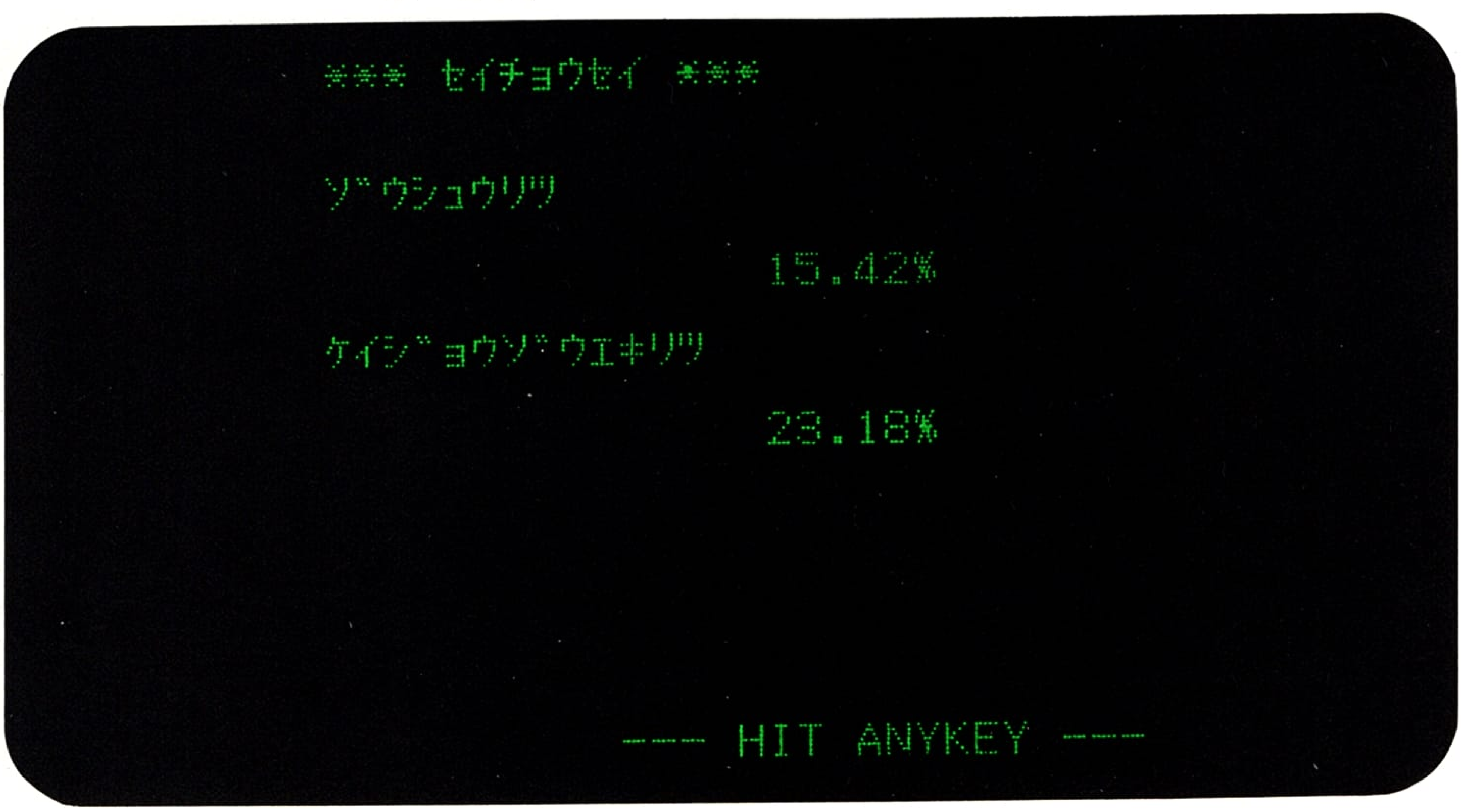


どれかキーを押せば、メニュー画面にもどります。

成長性の比率表示

成長性の比率表示

メニュー画面において、4 ⏎とキーインすれば、成長性の比率を表示します。



どれかキーを押します。メニュー画面にもどります。

ゴクロウサマ

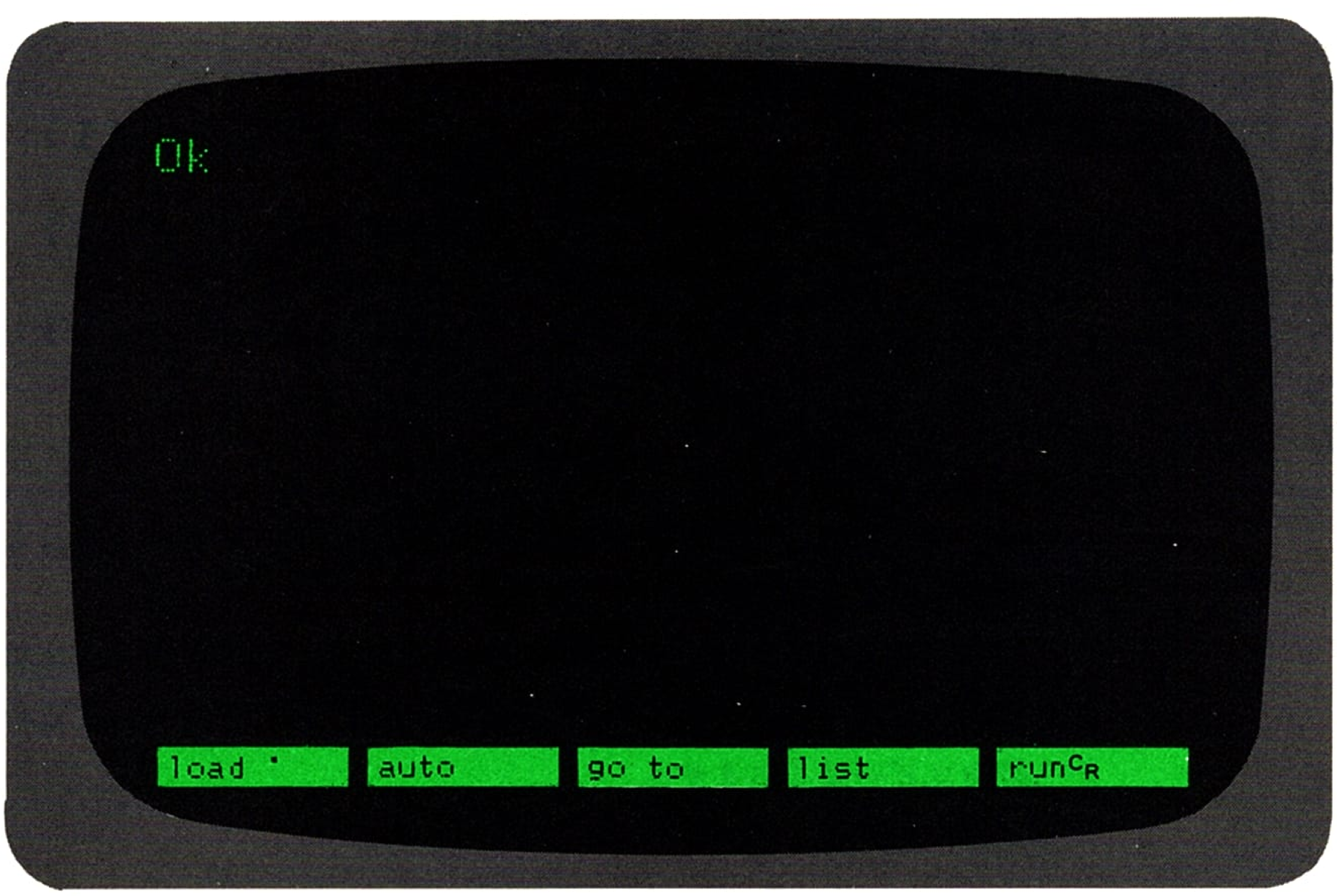
メニュー画面において、5 ↵とキーインすれば、次の画面になります。



ここで、どのキーでもいいから押せば、次の画面になり、経営分析プログラムの実行が終わります。

経営分析プログラムの実行の終了

経営分析プログラムの実行が終われば、画面の左上にＯｋが表示され、最下位行にはファンクションキーの内容が表示されます。



5・9 経営分析プログラムのまとめ

経営分析プログラムとは

経営分析プログラム

ここでの経営分析プログラムは、データをインプットすれば、比率を計算し、メニュー画面の選択により、収益性・安全性・生産性・成長性の比率をディスプレイの画面に表示をするプログラムです。

計算比率をプリンタに出力するには

収益性・安全性・生産性・成長性の比率をプリンタに出力するには、2つの方法があります。

画面コピー（スクリーンコピー）

第1の方法は、画面コピー（スクリーンコピー）をとるという方法です。画面コピーをとる方法には、さらに、2つの方法があります。

1つは、STOPキーを押して経営分析プログラムの実行をストップさせ、COPYキーを押すという方法です。

ＣＯＰＹ文

もう1つは、経営分析プログラムの中にＣＯＰＹ文を入れておき、経営分析プログラムの実行につれ画面コピーをとるという方法です。たとえば、収益性の比率を画面に表示したあと、画面コピーをとるのであれば、次のＣＯＰＹ文を経営分析プログラムに入れておけばよいのです。次のようにキーインしてみましょう。

```
2245 △ COPY ⏎   ← 画面コピーをとる
```

収益性の比率表示のプログラムは、次のようになります。

```
2180 PRINT "*** ｼｭｳｴｷｾｲ ***":PRINT
2190 PRINT :PRINT C$(1):PRINT
2200 PRINT USING "                    ###.##%  ";D(1)
2210 PRINT :PRINT C$(2):PRINT
2220 PRINT USING "                    ###.##%  ";D(2)
2230 PRINT :PRINT C$(3):PRINT
2240 PRINT USING "                    ###.##ｶｲ";D(3)
2245 COPY   ← ＣＯＰＹ文
2250 GOTO 2490
```

それでは、このＣＯＰＹ文を追加した経営分析プログラムを実行して、画面コピーをとってみてください。

ただし、あなたのプリンタが、ＰＣ-８８０１のＣＯＰＹ機能が実現できるプリンタ（たとえば、ＰＣ-８８２１やＰＣ-８８２２）でないと、画面コピーはとれません。

LPRINT文
LPRINT△USING文

第２の方法は、経営分析プログラムの中にＬＰＲＩＮＴ文やＬＰＲＩＮＴ　ＵＳＩＮＧ文を入れておき、経営分析プログラムの実行につれプリンタに出力するという方法です。

ＰＲＩＮＴ文やＰＲＩＮＴ　ＵＳＩＮＧ文がディスプレイに情報を出力するのにたいして、ＬＰＲＩＮＴ文やＬＰＲＩＮＴ　ＵＳＩＮＧ文はプリンタに情報を出力します。しかし、文（ステートメント）の形は、ＰＲＩＮＴやＰＲＩＮＴ　ＵＳＩＮＧというキーワードの前にＬがあるかないかのちがいです。

そこで、経営分析プログラムの中にＬＰＲＩＮＴ文やＬＰＲＩＮＴ　ＵＳＩＮＧ文を入れるのは簡単です。たとえば、収益性の比率表示のプログラムの中にＬＰＲＩＮＴ文やＬＰＲＩＮＴ　ＵＳＩＮＧ文を入れるには、次のようにすればよいのです。

LISTコマンド

まず、ＬＩＳＴコマンドを使って、収益性の比率表示のプログラムを画面に表示します。次のようにキーインしましょう。

f·4 2180-2250 ⏎ ← 行番号2180－2250の文を画面表示

画面には、次のように表示されます。

```
list 2180-2250
2180 PRINT "*** シュウエキセイ ***":PRINT
2190 PRINT :PRINT C$(1):PRINT
2200 PRINT USING "                ###.##%  ";D(1)
2210 PRINT :PRINT C$(2):PRINT
2220 PRINT USING "                ###.##%  ";D(2)
2230 PRINT :PRINT C$(3):PRINT
2240 PRINT USING "                ###.##カイ";D(3)
2250 GOTO 2490
Ok
■
```

収益性の比率表示のプログラム

カーソルはここでブリンク（点滅）

ここで、カーソルを動かし、ＰＲＩＮＴ文やＰＲＩＮＴ　ＵＳＩＮＧ文のある行番号の下１桁を０から５に変え、ＰＲＩＮＴやＰＲＩＮＴ　ＵＳＩＮＧというキーワードの前にＬを入れ、ＬＰＲＩＮＴやＬＰＲＩＴ　ＵＳＩＮＧに変えます。各行について、修正がすんだら、必ず、⏎キーを押します。

修正がすんだら、CLRキーを押して画面をクリアし、再度、収益性の比率表示のプログラムを画面に表示します。次のようにキーインします。

f·4 2180-2250 ⏎ ← 行番号2180－2250の文を画面表示

画面には、次のように表示されます。

```
list 2180-2250
2180 PRINT "※※※ ｼｭｳｴｷｾｲ ※※※":PRINT
2185 LPRINT "※※※ ｼｭｳｴｷｾｲ ※※※":LPRINT
2190 PRINT :PRINT C$(1):PRINT
2195 LPRINT :LPRINT C$(1):LPRINT
2200 PRINT USING "                   ###.##%  ";D(1)
2205 LPRINT USING "                   ###.##%  ";D(1)
2210 PRINT :PRINT C$(2):PRINT
2215 LPRINT :LPRINT C$(2):LPRINT
2220 PRINT USING "                   ###.##%  ";D(2)
2225 LPRINT USING "                   ###.##%  ";D(2)
2230 PRINT :PRINT C$(3):PRINT
2235 LPRINT :LPRINT C$(3):LPRINT
2240 PRINT USING "                   ###.##ｶｲ";D(3)
2245 LPRINT USING "                   ###.##ｶｲ";D(3)
2250 GOTO 2490
Ok
```

それでは、ＬＰＲＩＮＴ文とＬＰＲＩＮＴ　ＵＳＩＮＧ文を追加した経営分析プログラムを実行して、プリンタに収益性の比率を出力してみてください。

さらに、安全性・生産性・成長性の比率も、プリンタに出力するようにくふうしてみてください。

経営分析プログラムのセーブ

経営分析プログラムが完成したら、電源をＯＦＦにしてしまわないうちに、カセットテープあるいはディスクにセーブしておきましょう。

より良い経営分析プログラムにするには

経営分析

単に収益性・安全性・生産性・成長性の比率を計算しただけでは、経営分析はできません。経営分析をするには、自社前期比率との期間比較、業界平均値との比較、ライバル会社との比較、業界トップ会社との比較など、比較分析をする必要があります。いままでの経営分析プログラムでも、そのつど、自社前期データ、業界平均値データ、ライバル会社データ、業界トップ会社データなどをキーインして、何回か実行してやれば、それらの結果を比較することができます。しかし、より良い経営分析プログラムにするには、１本のプログラムで比較分析のできるプログラムにすべきです。ぜひ、挑戦してみてください。

第**6**章

設備投資計画プログラム

6・1　設備投資計画プログラムのフローチャート

設備投資計画プログラム

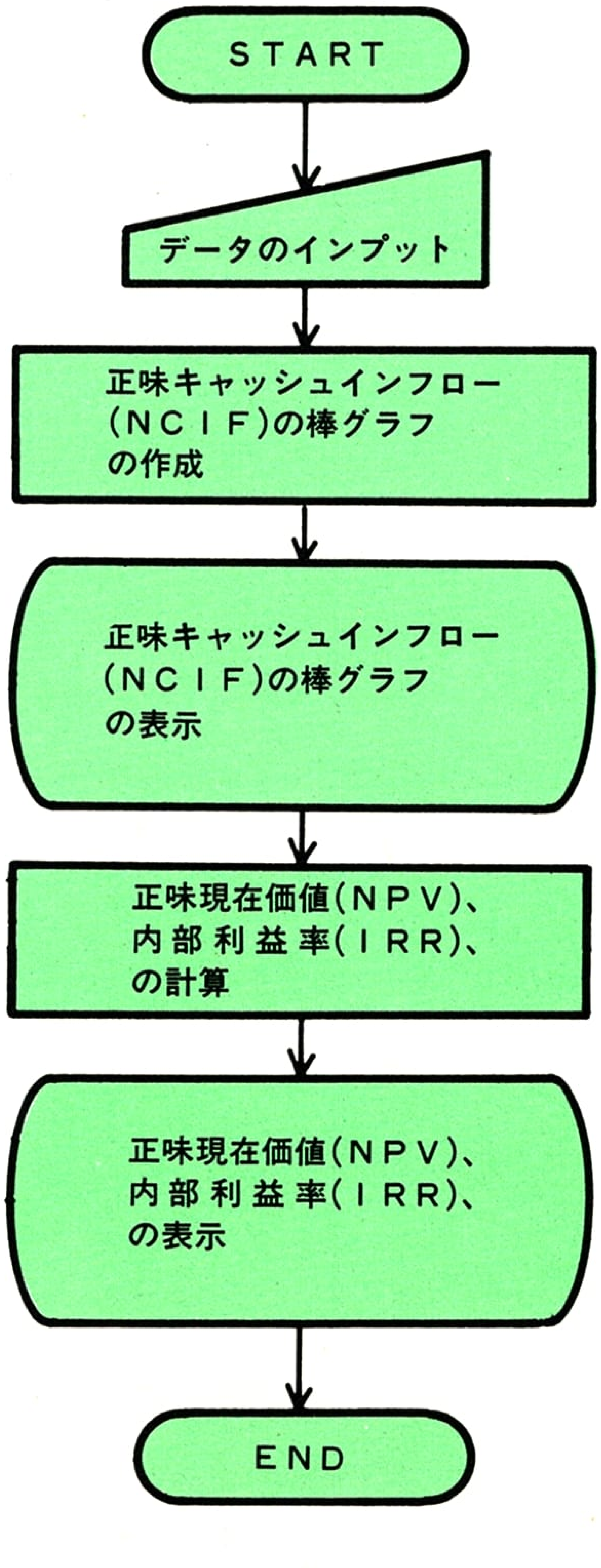
設備投資計画プログラムとは

ここでいう設備投資計画プログラムとは、設備投資計画案のデータ（投資額、耐用年数、正味キャッシュインフロー〈NetCash In Flow，ＮＣＩＦ〉、割引率）にもとづいて、ＮＣＩＦの棒グラフを作成し、正味現在価値（Net Present Value,ＮＰＶ）と内部利益率（Internal Rate of Return,ＩＲＲ）を計算するプログラムをいいます。

設備投資計画プログラムのフローチャート

設備投資計画プログラムのフローチャート

設備投資計画プログラムのフローチャートは、次のとおりです。



それでは、このフローチャートにしたがって、設備投資計画プログラムの全体的な流れを見てみましょう。

まず、設備投資計画プログラムをスタートさせます。

そして、データをインプットします。ここでは、インプット項目は次の４項目です。

(1)　投資額……ＣＩ

(2)　耐用年数……ｎ

(3)　毎年度末の正味キャッシュインフロー……ＮＣＩＦ（ｔ）

〈ｔ＝１，２，……，ｎ〉

(4)　割引率……ｒ

ただし、現在時点（第０年度末）に設備投資がなされ、第１年度以降毎年度末に正味キャッシュインフローが発生すると仮定します。

次に、毎年度末の正味キャッシュインフローの棒グラフを作成し、画面に表示します。

続いて、正味現在価値（ＮＰＶ）、内部利益率（ＩＲＲ）を計算し、画面に表示します。

そして、設備投資計画プログラムの実行を終了します。

正味現在価値（ＮＰＶ）

なお、正味現在価値（ＮＰＶ）は、設備投資にともなう正味キャッシュインフローの現在価値合計から投資額を差し引いたものです。したがって、正味現在価値（ＮＰＶ）の計算式は次のとおりです。

$$NPV=\frac{NCIF(1)}{(1+r)}+\frac{NCIF(2)}{(1+r)^2}+\frac{NCIF(3)}{(1+r)^3}+\cdots+\frac{NCIF(n)}{(1+r)^n}-CI$$

内部利益率（ＩＲＲ）

また、内部利益率（ＩＲＲ）は、正味現在価値（ＮＰＶ）をゼロにする割引率、すなわち正味キャッシュインフローの現在価値合計と投資額とを等しくする割引率です。したがって、内部利益率（ＩＲＲ）は、次の式を満足するｒです。

$$\frac{NCIF(1)}{(1+r)}+\frac{NCIF(2)}{(1+r)^2}+\frac{NCIF(3)}{(1+r)^3}+\cdots+\frac{NCIF(n)}{(1+r)^n}=CI$$

こうした正味現在価値（ＮＰＶ）や内部利益率（ＩＲＲ）の計算のしかたなど設備投資計画については、次のものを参照してください。

拙稿、「設備投資計画」、宮川公男編『経営学』青林書院新社、昭和56年、第５章に所収、139－167頁。

それでは、このフローチャートにしたがって、設備投資計画プログラムを作成してみましょう。

6・2 スタート部分のプログラム

スタート部分のプログラムのリスト

設備投資計画プログラムのスタート部分のプログラムのリストは、次のとおりです。

```
10 REM ********** ｾﾂﾋﾞﾄｳｼｹｲｶｸ **********
20 REM ***** START *****
30 WIDTH 40,25
40 CONSOLE 0,25,0,1
50 COLOR 4
60 A31$=STRING$(31,"*")          ← STRING$関数
70 A8$=STRING$(8,"*")            ← STRING$関数
80 FOR I=1 TO 5
90 PRINT
100 NEXT I
110 FOR I=1 TO 5
120 PRINT SPC(4);A31$            ← SPC関数
130 NEXT I
140 FOR I=1 TO 2
150 PRINT SPC(4);A8$;SPC(15);A8$
160 NEXT I
170 PRINT SPC(4);A8$;SPC(2);"ｾﾂﾋﾞﾄｳｼｹｲｶｸ";SPC(2);A8$
180 FOR I=1 TO 2
190 PRINT SPC(4);A8$;SPC(15);A8$
200 NEXT I
210 FOR I=1 TO 5
220 PRINT SPC(4);A31$
230 NEXT I
240 FOR I=1 TO 1000              ← 時間かせぎ
250 NEXT I                       ←
```

このスタート部分のプログラムのリストの中で、新しく出てきたものは、ＳＴＲＩＮＧ＄関数とＳＰＣ関数です。

ＳＴＲＩＮＧ＄関数

行番号60、70の代入文を見てください。次の形をしたＳＴＲＩＮＧ＄（ストリングドル）関数が使われています。

```
STRING$（数値 , 文字）
```

このＳＴＲＩＮＧ＄関数によって、指定した数値の数だけ指定した文字を与えることができます。

したがって、行番号60の代入文を実行すると、31個の＊（アスタリスク）からなるストリング（文字列）が文字型変数Ａ31＄に代入されます。同じように、行番号70の代入文を実行すると８個の＊からなるストリングがＡ８＄に代入されます。

ＳＰＣ関数

行番号120のＰＲＩＮＴ文を見てください。次の形をしたＳＰＣ（エスピーシー）関数が使われています。

SPC（数値）

このＳＰＣ関数によって、ＰＲＩＮＴ文やＬＰＲＩＮＴ文の中で、指定の数だけスペース（空白）を出力することができます。

したがって、行番号120のＰＲＩＮＴ文を実行すると、スペースを４文字あけ、＊を31文字出力します。

ただし、行番号120のＰＲＩＮＴ文は、行番号110～130のＦＯＲ～ＮＥＸＴループの中にあります。そこで、このＰＲＩＮＴ文は、５回繰り返し実行されます。よろしいですね。

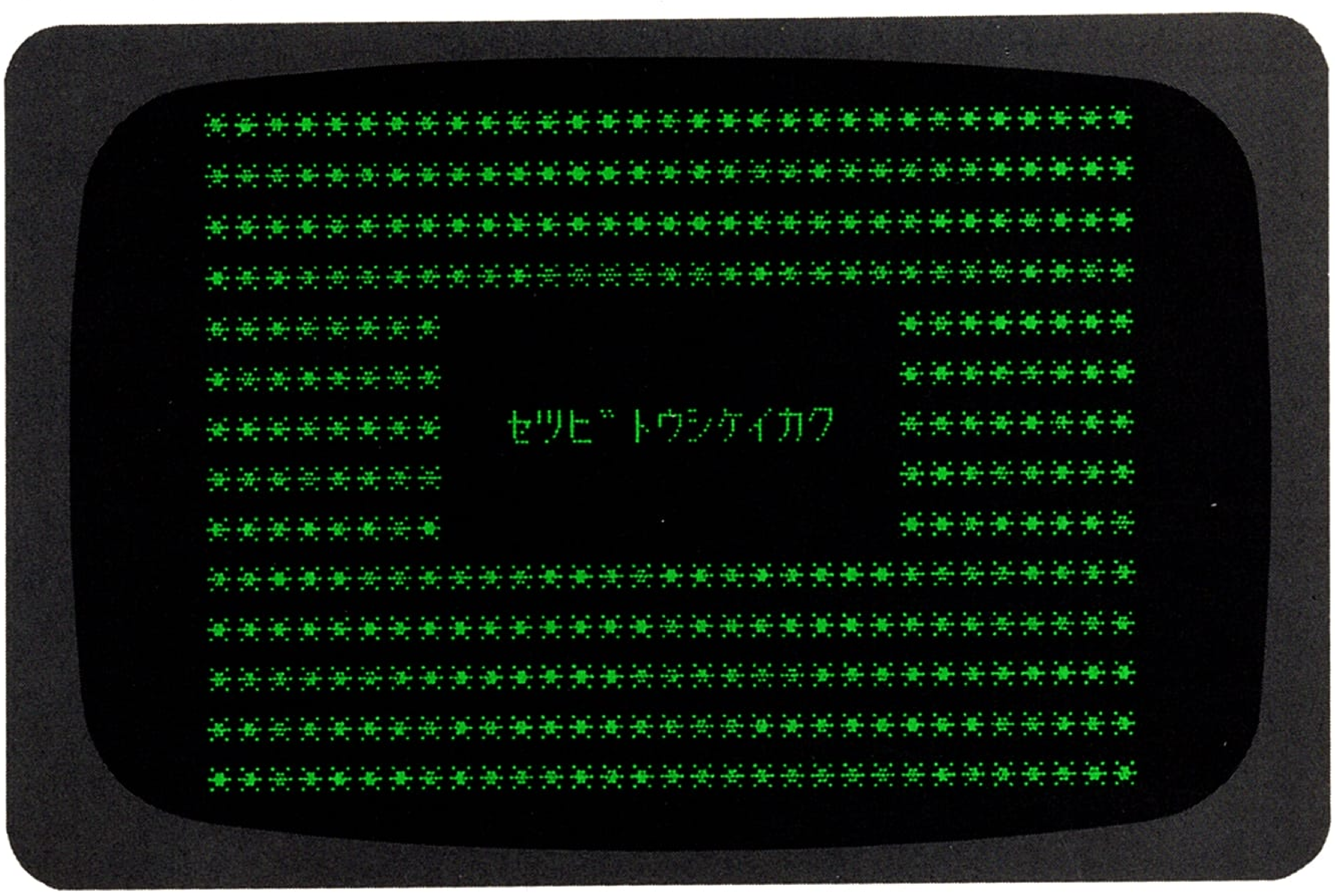
行番号140～160、180～200、210～230も同じようなＦＯＲ～ＮＥＸＴループになっていますね。

行番号170のＰＲＩＮＴ文の意味もわかりますね。

時間かせぎ

なお、行番号240～250のＦＯＲ～ＮＥＸＴループは、時間かせぎのためのものです。つまり、Ｉが１～1000になるまで時間かせぎをして、あまり早く次の行番号1000へ行かないようにしているわけです。

以上のスタート部分のプログラムを実行すると、次のように画面に表示されます。



6・3　データインプットのプログラム

データのインプット

データインプットのプログラムは、次のとおりです。

```
1000 REM ***** DATA ノ INPUT *****
1010 CLS
1020 COLOR 5
1030 M3$=STRING$(3,"-")                          ← STRING$関数
1040 PRINT M3$;" data ヲ input シテクダサイ ";M3$
1050 PRINT
1060 INPUT "トウシガク        = ";CI             ← INPUT文
1070 INPUT "タイヨウネンスウ    = ";N              ← INPUT文
1080 DIM NCIF(N)                                 ← DIM文
1090 PRINT "ショウミ キャッシュ インフロー : NCIF(T)"
1100 PRINT "  ネン(T)  NCIF( T )"
1110 FOR T=1 TO N
1120 PRINT SPC(4);T;SPC(2);"NCIF(";T;") = ";
1130 INPUT NCIF(T)                               ← INPUT文
1140 NEXT T
1150 INPUT "ワリビキリツ(%) = ";R                 ← INPUT文
```

ＳＴＲＩＮＧ＄関数

行番号1030の代入文にＳＴＲＩＮＧ＄関数が使われています。

したがって、行番号1030の代入文を実行すると、３個の－（マイナス）からなるストリングが文字型変数Ｍ３＄に代入されます。

行番号1060のＩＮＰＵＴ文を実行すると、投資額が変数ＣＩに代入されます。また、行番号1070のＩＮＰＵＴ文を実行すると、耐用年数が変数Ｎに代入されます。

そして、行番号1080のＤＩＭ文によって、正味キャッシュインフローの配列が宣言されます。配列変数ＮＣＩＦ（Ｎ）の添字Ｎは、耐用年数の値Ｎです。

次に、行番号1110～1140のＦＯＲ～ＮＥＸＴループの中の行番号1130のＩＮＰＵＴ文により、毎年度末の正味キャッシュインフローが配列変数ＮＣＩＦ（Ｔ）に代入されます。

また、行番号1150のＩＮＰＵＴ文により、割引率が変数Ｒに代入されます。

インプットデータの画面表示

インプットデータを見やすくするためのプログラムは、次のとおりです。

```
1200 REM *** input data ***
1210 CLS
1220 A3$=STRING$(3,"*")
1230 PRINT A3$;" input data ";A3$
1240 PRINT
1250 PRINT "( 1) トウシカ゛ク    = ";
1260 PRINT USING "###,###";CI
1270 PRINT "( 2) タイヨウネンスウ =   ";N;"ネン"
1280 FOR T=1 TO N
1290 T2=T+2
1300 PRINT USING "(##";T2;
1310 PRINT USING ") NCIF(##";T;
1320 PRINT USING ") = ###,###";NCIF(T)
1330 NEXT T
1340 T3=T2+1
1350 PRINT USING "(##";T3;
1360 PRINT ") ワリヒ゛キリツ  = ";R;"%"
1370 PRINT
```

代入文（1290、1340）
PRINT USING文（1260、1300、1310、1320、1350）

インプットデータを見やすくするため、行番号1260、1300、1310、1320、1350のＰＲＩＮＴ　ＵＳＩＮＧ文が使われています。

また、データの修正ができるようにするため、データに番号をつけておきます。行番号1290の代入文は、正味キャッシュインフローＮＣＩＦ（1）、ＮＣＩＦ（2）、…、ＮＣＩＦ（n）の番号を(3)、(4)、…、(n＋2)番とするため、年度をあらわす変数Ｔに２を加えたものを変数Ｔ２に代入します。同じように、割引率Ｒの番号を(n＋3)番とするため、変数Ｔ２（この値はn＋2になっている）に１を加えたものを変数Ｔ３に代入します。よろしいですね。

インプットデータの修正

インプットデータを修正するためのプログラムは、次のとおりです。

```
1400 REM *** data ノ シュウセイ ***
1410 PRINT " data ヲ シュウセイ シマスカ ?"
1420 PRINT "                  1 ";M3$;" スル"
1430 PRINT "                  0 ";M3$;" シナイ"
1440 PRINT "                              ";
```

```
1450 INPUT CD
1460 IF CD=1 GOTO 1490
1470 IF CD=0 GOTO 2000
1480 GOTO 1400
1490 PRINT
1500 PRINT "ﾅﾝﾊﾞﾝ ﾉ data ｦ ｼｭｳｾｲ ｼﾏｽｶ ";
1510 INPUT I
1520 ON I GOTO 1560,1580                  ← ON…GOTO文
1530 IF I>=3 AND I<=T2 GOTO 1600          ← IF…GOTO文
1540 IF I=T3 GOTO 1650                    ←
1550 GOTO 1490
1560 INPUT "( 1) ﾄｳｼｶﾞｸ    = ";CI
1570 GOTO 1200
1580 INPUT "( 2) ﾀｲﾖｳﾈﾝｽｳ = ";N
1590 GOTO 1200
1600 PRINT USING "(##";I;
1610 PRINT USING ") NCIF(##";I-2;
1620 PRINT ") = ";
1630 INPUT NCIF(I-2)
1640 GOTO 1200
1650 PRINT USING "(##";T3;
1660 INPUT ") ﾜﾘﾋﾞｷﾘﾂ  = ";R
1670 GOTO 1200
```

ON…GOTO文

行番号1520のＯＮ（オン）…ＧＯＴＯ（ゴウトゥ）文を見てください。ＯＮ…ＧＯＴＯ文は、指定されたいくつかの行番号の行に飛ぶためのものです。次の形になっています。

ON △ 変数 △ GOTO △ 行番号，行番号，…

変数の値が１なら１番目の行番号の行に飛び、変数の値が２なら２番目の行番号の行に飛びます。

したがって、行番号1520のＯＮ…ＧＯＴＯ文を実行すると、変数Ｉの値（修正すべきデータ番号）が１なら行番号1560のＩＮＰＵＴ文に飛び、Ｉの値が２なら行番号1580のＩＮＰＵＴ文に飛びます。

ＩＦ…ＧＯＴＯ文

次に、行番号1530のＩＦ…ＧＯＴＯ文を見てください。Ｉ＞＝３は、Ｉ≧３、つまりＩの値が３以上という意味です。同じように、Ｉ＜＝Ｔ２は、Ｉ≦Ｔ２、つまりＩの値がＴ２以下という意味です。また、Ｉ＞＝３　ＡＮＤ　Ｉ＜＝Ｔ２は、Ｉ≧３かつＩ≦Ｔ２、つまりＩの値が３以上かつＴ２以下という意味です。そこで、行番号1530のＩＦ…ＧＯＴＯ文を実行すると、Ｉの値が３以上かつＴ２以下なら行番号1600のＰＲＩＮＴ　ＵＳＩＮＧ文へ飛び、正味キャッシュインフロー

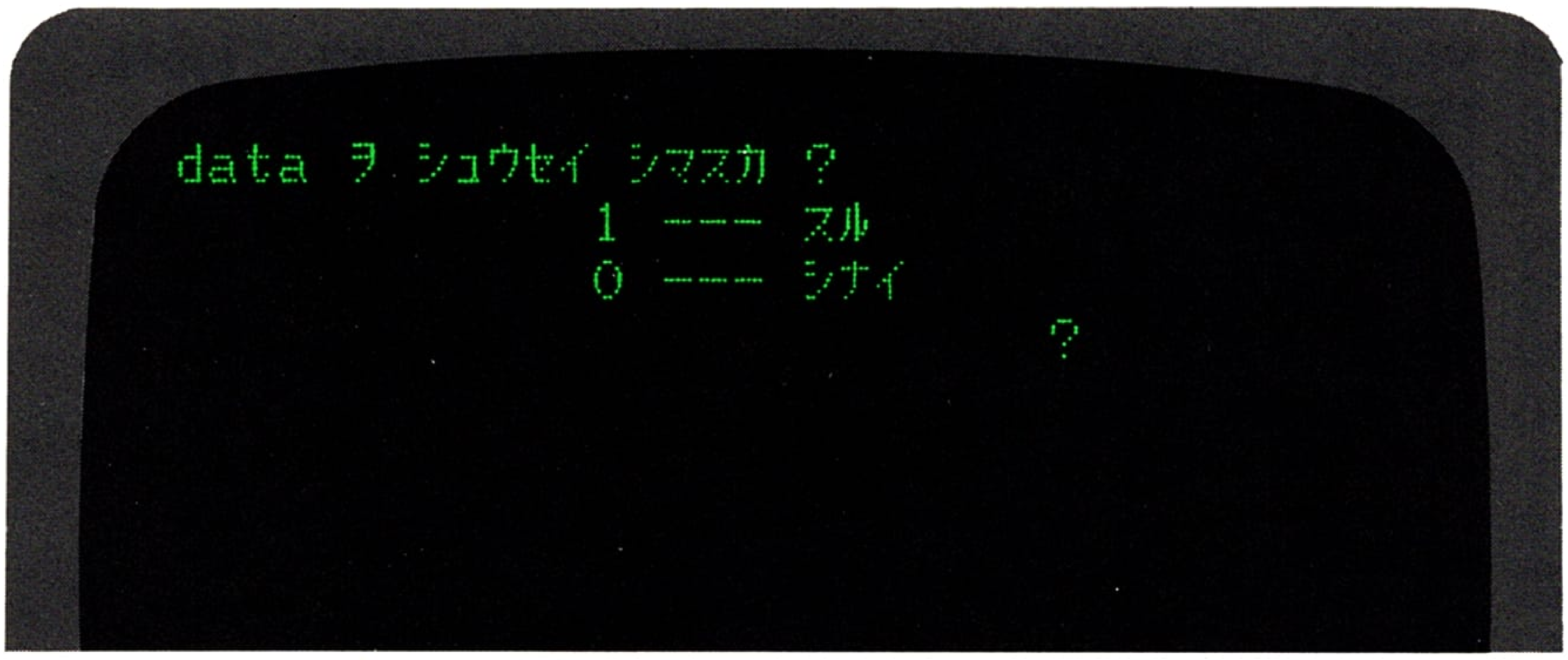
ＮＣＩＦ（Ｉ－２）をインプットします。

なお、Ｉ－２、つまり修正すべきデータ番号から２を引けば、年度が計算されるので、ＮＣＩＦ（Ｉ－２）は修正すべき年度の正味キャッシュインフローを意味します。

そして、行番号1540のＩＦ…ＧＯＴＯ文も同じように、Ｉ＝Ｔ３なら行番号1650のＰＲＩＮＴ　ＵＳＩＮＧ文へ飛び、割引率Ｒをインプットします。

以上のインプットデータの修正プログラムを実行すると、次のようになります。

まず、行番号1410～1440のＰＲＩＮＴ文で、データを修正するか否かのメッセージが画面に表示されます。



そして、行番号1450のＩＮＰＵＴ文です。データを修正するのであれば、１をキーインし、修正しないのであれば０をキーインします。１あるいは０は、変数ＣＤに代入されます。

次に、データを修正する（ＣＤ＝１）であれば、行番号1490へ飛んできて、何番のデータを修正するかをキーインさせ、その値を変数Ｉに代入します。そして、Ｉ番目のインプット項目名を画面に表示して、正しいデータをキーインさせます。

こうして、データの修正が終わったら、行番号1570、1590、1640、あるいは1670のＧＯＴＯ文により、行番号1200へもどり、再度インプットデータを画面に表示します。そして、データを修正するか否かのメッセージが画面に表示されるので、さらに修正したいデータがあれば１をキーインすればよいし、すべてのデータが正しくキーインできていれば０をキーインすればいいのです。

すべてのデータが正しくキーインできていれば、行番号1470のＩＦ…ＧＯＴＯ文により、行番号2000へ飛びます。

6・4　棒グラフの作成

ＮＣＩＦの棒グラフの作成

ここで、正味キャッシュインフローＮＣＩＦの棒グラフを作成し、画面に表示してみましょう。

棒グラフ作成のプログラム

棒グラフ作成のプログラムは、次のとおりです。

```
2000 REM ***** NCIF ﾉ ﾎﾞｰｸﾞﾗﾌ *****
2010 CLS
2020 COLOR 2
2030 PRINT SPC(4);A8$;" NCIF ﾉ ﾎﾞｰｸﾞﾗﾌ ";A8$
2040 PRINT : PRINT
2050 MAX=0
2060 FOR T=1 TO N
2070 IF MAX<NCIF(T) THEN MAX=NCIF(T)
2080 NEXT T
2090 SCALE=MAX/20
2100 FOR T=1 TO N
2110 COLOR 6
2120 PRINT USING "NCIF(##";T;
2130 PRINT ") ";
2140 PLOT=CINT(NCIF(T)/SCALE)
2150 FOR P=1 TO PLOT
2160 COLOR 4
2170 PRINT "■";
2180 NEXT P
2190 COLOR 5
2200 PRINT USING "   ###,###";NCIF(T)
2210 PRINT
2220 NEXT T
2230 GOSUB 5000
```

仮の最大値を０とする

最大値を求める

グラフ上のスケーリング単位を求める

プロット回数を求める

■をプロットする

ＧＯＳＵＢ文

行番号2050の代入文を見てください。毎年度の正味キャッシュインフローのうち最大なものをＭＡＸとします。ここでは、ＭＡＸに０を代入します。つまり、仮の最大値を０とします。

仮の最大値

次に、行番号2060～2080のＦＯＲ～ＮＥＸＴループで、正味キャッシュインフローの最大値を求めます。行番号2070のＩＦ…ＴＨＥＮ文

最大値を求める ＩＦ…ＴＨＥＮ文

で、MAXの値がNCIF（T）の値より小さければ、NCIF（T）の値をMAXとします。MAXの値がNCIF（T）の値より大きいか等しければ、MAXと次年度のNCIFとを比較します。

そして、行番号2090の代入文で、正味キャッシュインフローの最大値MAXを20で割って、グラフ上のスケーリング単位を求めます。なお、棒グラフを画面に表示するときに、グラフィックシンボル■を何回か横方向にプロット（出力）して棒グラフとします。そのさい、MAXの値については、■を20回プロットすることにします。MAXの値について、■を25回プロットするのであれば、MAXを25で割ればよいのです。

スケーリング単位

■をプロット

行番号2140の代入文でプロット回数を計算します。CINT（シーイント）関数を使って、NCIF（T）をSCALEで割った値を整数値に変換した値をプロット回数とします。CINT関数は、次の形をしています。

CINT関数

```
CINT（数式）
```

CINT関数によって、数式の値の小数点以下を四捨五入して整数に変換した値がえられます。

行番号2150～2180のFOR～NEXTループによって、■をPLOTの値だけプロットします。

なお、グラフィックシンボル■をキーインするには、GRAPH キーを押しながら ネ キーを押します。■のかわりに、＊や♥などを使った棒グラフにしてもかまいません。あとでためしてみてください。

ここで、行番号2230のGOSUB（ゴーサブ）文を見てください。このGOSUB文は、後に出てくるRETUN（リターン）文と対（つい）あるいはペアになっていて、GOSUB～RETURN文と呼ばれます。次の形をしています。

GOSUB文

RETURN文

GOSUB~RETURN文

```
GOSUB △ 行番号
  ⟆
RETURN
```

GOSUB文で指定した行番号から始まるサブルーチンプログラムをGOSUB文によってコールし（呼び出し），サブルーチンプログラム内のRETURN文によって、GOSUB文の次の文へリターンします（もどります)。

サブルーチンプログラムについては、「6・6　エンド部分とサブルーチンのプログラム」のところで説明します。

6·5 NPV, IRRの計算プログラム

まず、次のようにします。

```
3000 REM ***** NPV,IRR ノ ケイサン *****
3010 CLS
3020 COLOR 6
3030 PRINT M3$;" ケイサン ケッカ ";M3$
3040 PRINT
```

NPVの計算プログラム

正味現在価値（NPV）計算プログラム

正味現在価値（ＮＰＶ）の計算プログラムは、次のとおりです。

```
3100 REM *** NPV ノ ケイサン ***
3110 TPV=0
3120 FOR T=1 TO N
3130 TPV=TPV+NCIF(T)/(1+R/100)^T    ← TPVの計算
3140 NEXT T
3150 NPV=TPV-CI                     ← NPVの計算
3160 PRINT "NPV = ";
3170 PRINT USING "###,###.###";NPV
3180 PRINT
```

ＮＰＶの計算式については、「６・１　設備投資計画プログラム」のところを見てください。

なお、ＴＰＶは、設備投資にともなう正味キャッシュインフローの現在価値合計です。

IRRの計算プログラム

内部利益率（IRR）の計算プログラム

内部利益率（ＩＲＲ）の計算プログラムは、次のとおりです。

```
3200 REM *** IRR ノ ケイサン ***
3210 RR=0                           ← RRの初期値を0とする
3220 TPV=0
3230 FOR T=1 TO N
3240 TPV=TPV+NCIF(T)/(1+RR/100)^T   ← TPVの計算
3250 NEXT T
3260 NPV=TPV-CI                     ← NPVの計算
3270 IF NPV=0 GOTO 3430
3280 IF NPV<0 GOTO 3310             ← NPVの値はマイナスにかわったか？
3290 RR=RR+10                       ← RRを10だけ増加させる
```

```
3300 GOTO 3220
3310 RR=RR-1                                   ← RRを1だけ減少させる
3320 TPV=0
3330 FOR T=1 TO N
3340 TPV=TPV+NCIF(T)/(1+RR/100)^T             ← TPVの計算
3350 NEXT T
3360 NPV=TPV-CI                                ← NPVの計算
3370 IF NPV=0 GOTO 3430
3380 IF NPV>0 GOTO 3410                        ← NPVの値はプラスにかわったか？
3390 NPV1=NPV
3400 GOTO 3310
3410 IRR=RR+NPV/(NPV-NPV1)                     ← 補間法
3420 GOTO 3440
3430 IRR=RR
3440 PRINT "IRR = ";IRR;"%"
3450 GOSUB 5000                                ← GOSUB文
```

ＩＲＲの計算式については、「6・1　設備投資計画プログラム」のところを見てください。

ここでは、ＩＲＲを次のように計算することにします。

まず、0％から10％きざみで利益率ＲＲを増加させ、そのつどＲＲで毎年度の正味キャッシュインフローＮＣＩＦ（Ｔ）を割り引いて現在価値合計ＴＰＶを計算し、正味現在価値ＮＰＶを求めます。そして、ＮＰＶがプラスの値からマイナスの値にかわったら、こんどはＲＲを1％きざみで減少させ、ＮＰＶがマイナスの値からプラスの値にかわる直前のＲＲとＮＰＶがプラスの値にかわった直後のＲＲとから、補間法を使って、内部利益率ＩＲＲを計算します。なお、ＮＰＶがゼロになれば、そのときのＲＲが求めるＩＲＲです。

ＩＲＲの計算は、数値例を考えたほうがわかりやすいと思います。数値例については、「6・8　設備投資計画プログラムのまとめ」を見てください。

いちど、手計算でＩＲＲを計算してみてください。ＩＲＲの計算プロセスがよくわかるはずです。しかし、手計算でＩＲＲを計算しようとすると、電卓を使ったとしても、かなりの時間がかかります。パソコンを利用すれば、たやすく計算できます。パソコンの有効性がよくわかります。

6・6　エンド部分とサブルーチンのプログラム

エンド部分のプログラムのリスト

設備投資計画プログラムのエンド部分のプログラムのリストは、次のとおりです。

```
4000 REM ***** END *****
4010 CLS
4020 COLOR 4
4030 PRINT M3$;" オワリ ";M3$
4040 PRINT
4050 PRINT " オツカレサマデ゛シタ "
4060 PRINT
4070 PRINT " オワリデ゛ス "
4080 GOSUB 5000          ← GOSUB文
4090 CLS
4100 WIDTH 80,20
4110 CONSOLE 0,20,1,0
4120 END
```

サブルーチンプログラムのリスト

サブルーチンプログラムのリストは、次のとおりです。

```
5000 REM ***** SUB (HIT ANYKEY) *****
5010 COLOR 1
5020 LOCATE 10,23
5030 PRINT M3$;" HIT ANYKEY ";M3$
5040 Z$=INKEY$
5050 IF Z$="" GOTO 5020
5060 BEEP
5070 RETURN          ← RETURN文
```

メインプログラム
サブルーチンプログラム

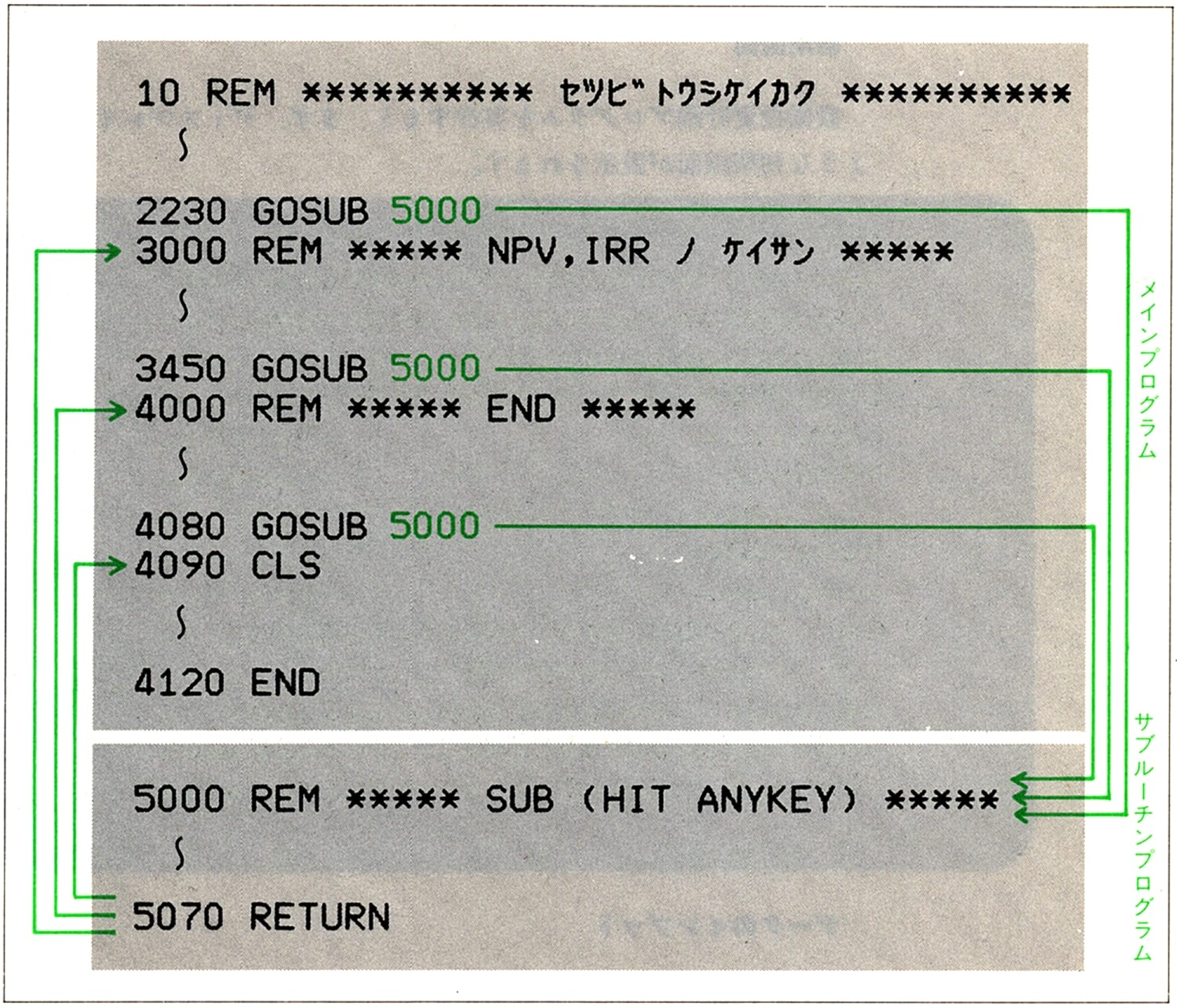
設備投資計画プログラムのうち、スタート部分からエンド部分までのプログラム、つまり行番号10～4120のプログラムをメインプログラムといい、行番号5000～5070のプログラムをサブルーチンプログラムといいます。

RETURN文
GOSUB文

サブルーチンプログラムは、他のプログラムから独立したプログラムです。また、サブルーチンプログラムは、ＲＥＴＵＲＮ文で終わります。こうしたサブルーチンプログラムは、ＧＯＳＵＢ文によって、

いつでも何度でもコールする（呼び出す）ことができます。

メインプログラムの行番号2230、3450、4080のＧＯＳＵＢ文によって、行番号5000から始まるサブルーチンプログラムがコールされ、画面に「―――　ＨＩＴ　ＡＮＹＫＥＹ　―――」のメッセージが表示されます。そして、どれかキーを押すと、行番号5070のＲＥＴＵＲＮ文によって、メインプログラムのＧＯＳＵＢ文の次の文へリターンします（もどります）。



設備投資計画プログラムのリスト

ＬＩＳＴコマンド

これで、設備投資計画プログラムも終わりです。ＬＬＩＳＴコマンドを使って、設備投資計画プログラムのリストをプリンタに出力してみてください。

ＬＩＳＴコマンドを使って、設備投資計画プログラムを画面に表示してみましょう。

ＬＬＩＳＴコマンド

また、ＬＬＩＳＴコマンドを使って、設備投資計画プログラムのリストをプリンタに出力してみてください。

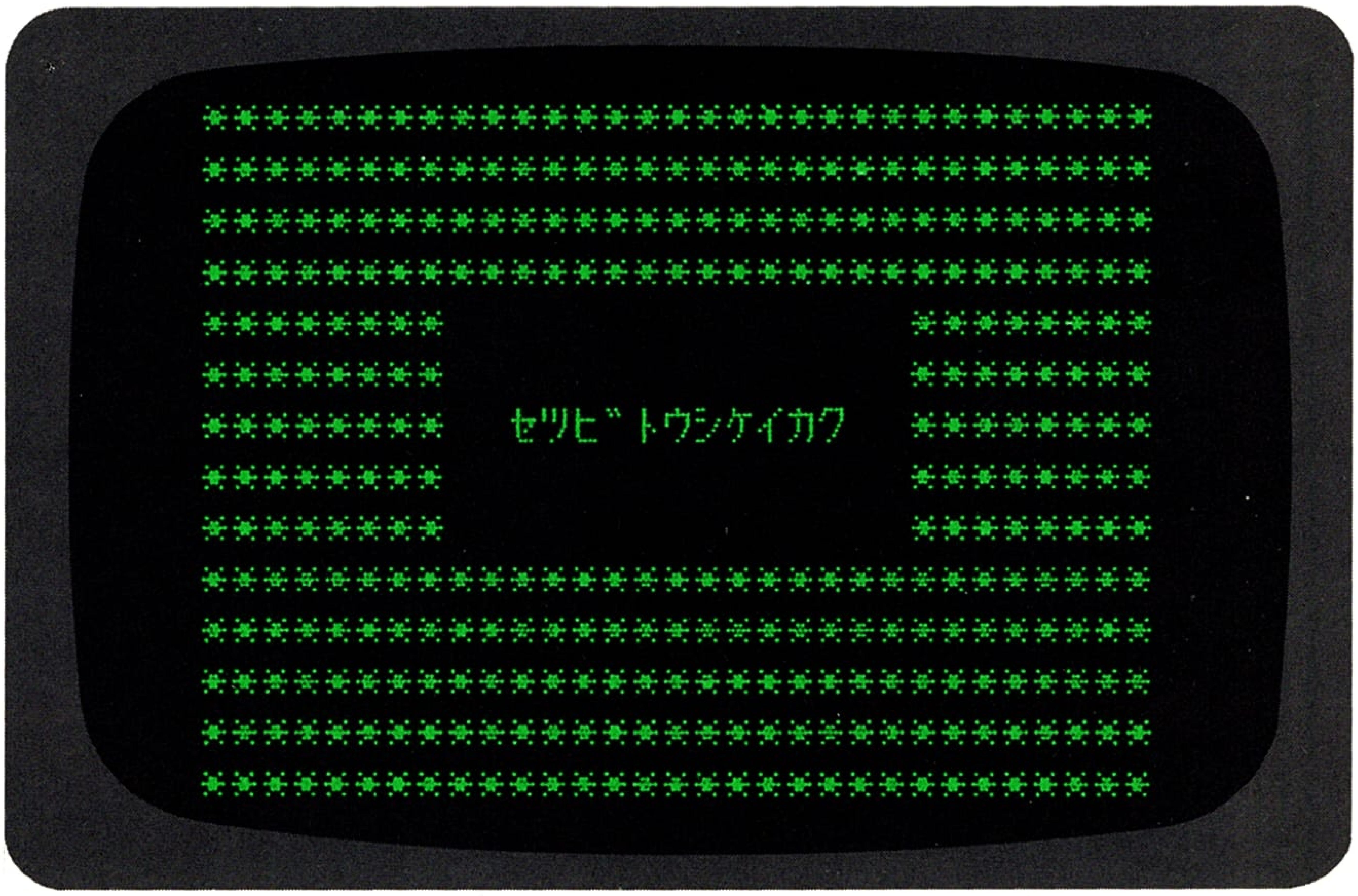
6・7　設備投資計画プログラムの実行

RUN

ＲＵＮコマンドを使って、設備投資計画プログラムを実行してみましょう。

初期画面

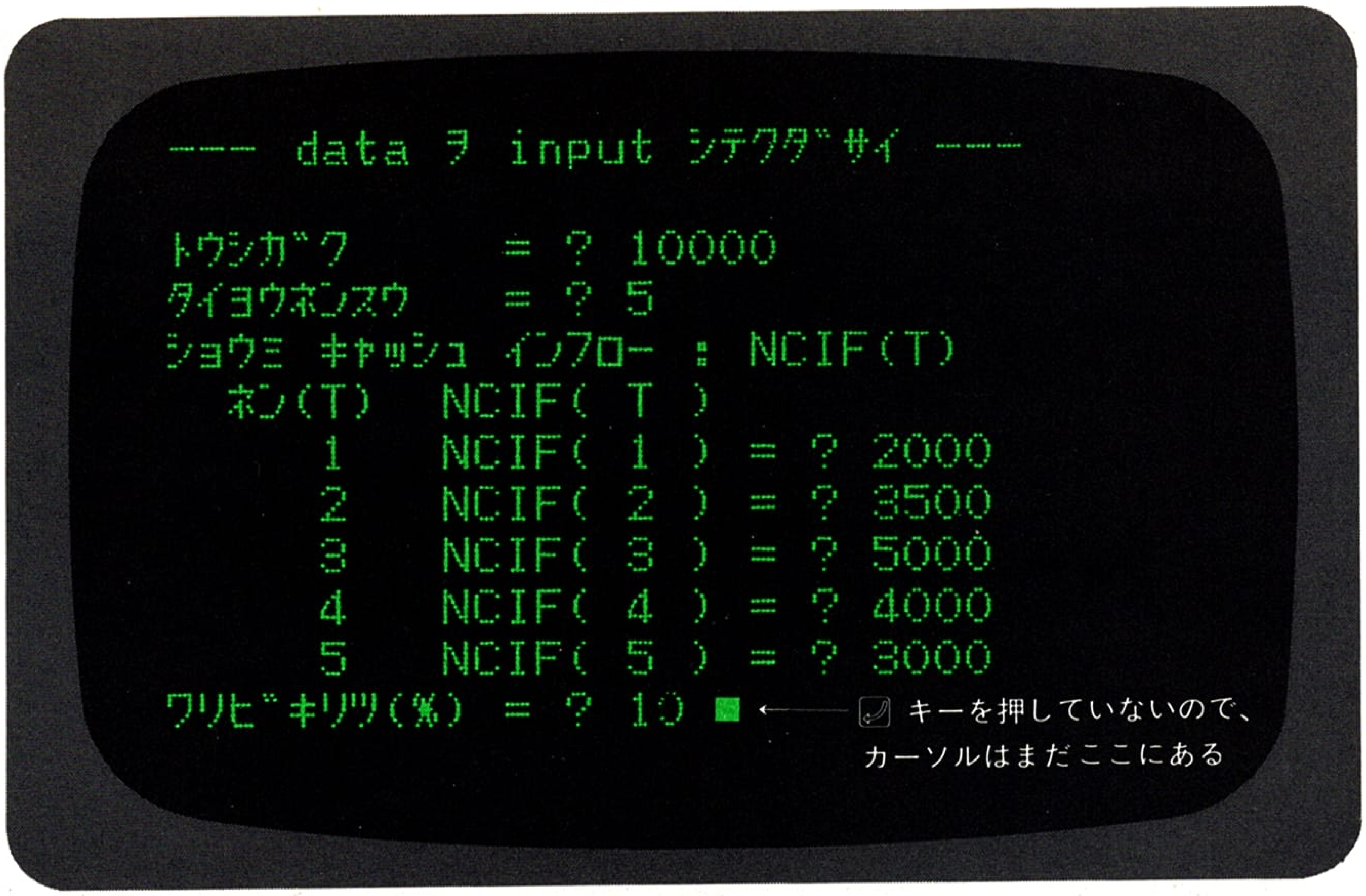
設備投資計画プログラムを実行すると、まず、ディスプレイに次のような初期画面が表示されます。



データのインプット

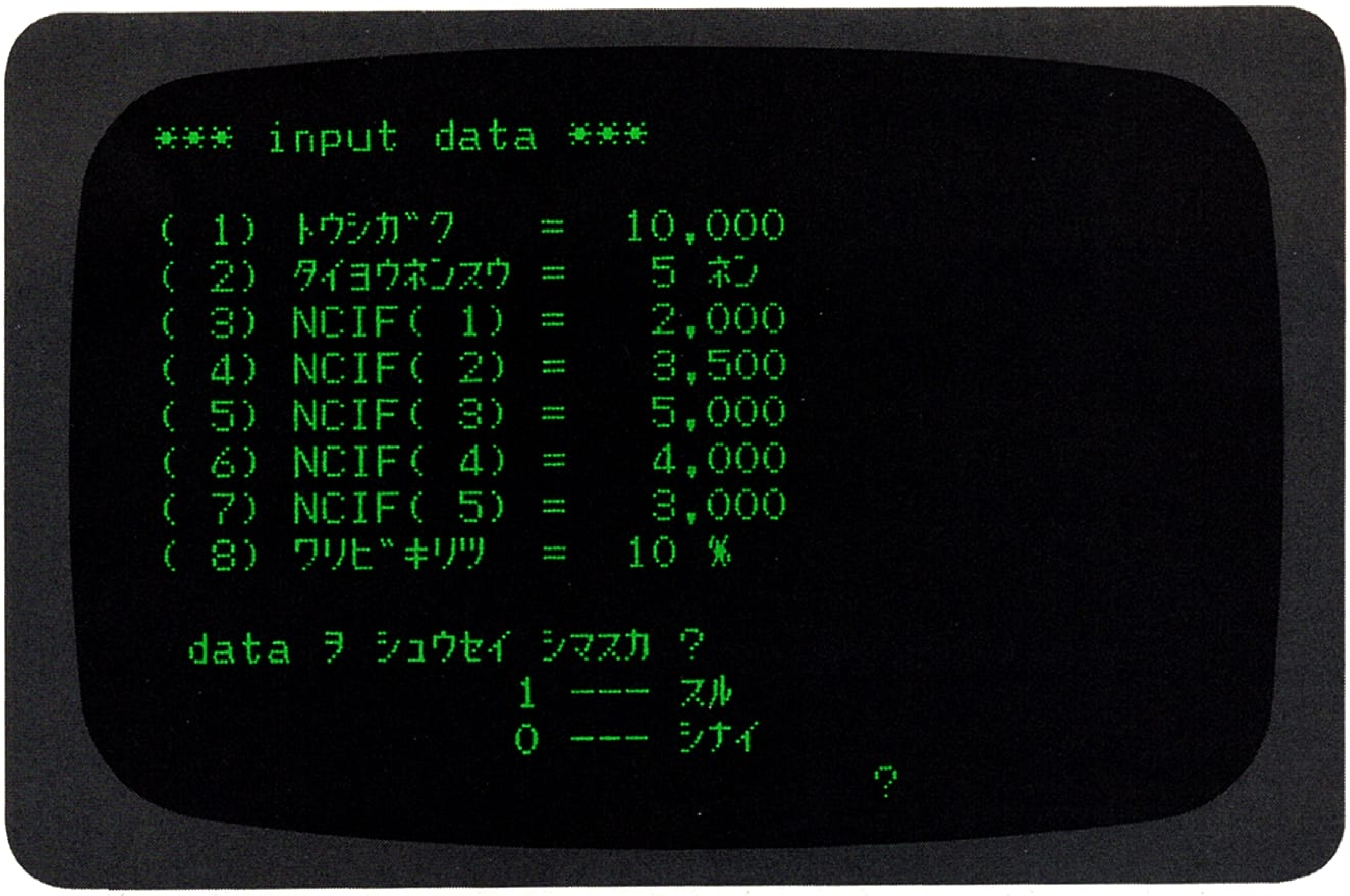
次に、画面のメッセージにしたがって、データをキーインします。

投 資 額＝10,000万円
耐用年数＝５年
毎年度末の正味キャッシュインフロー：
　ＮＣＩＦ(1)＝2,000万円
　ＮＣＩＦ(2)＝3,500
　ＮＣＩＦ(3)＝5,000
　ＮＣＩＦ(4)＝4,000
　ＮＣＩＦ(5)＝3,000
割 引 料＝10%



インプットデータの見やすい画面表示

ここで、⏎キーを押すと、インプットデータが画面に見やすく表示されます。



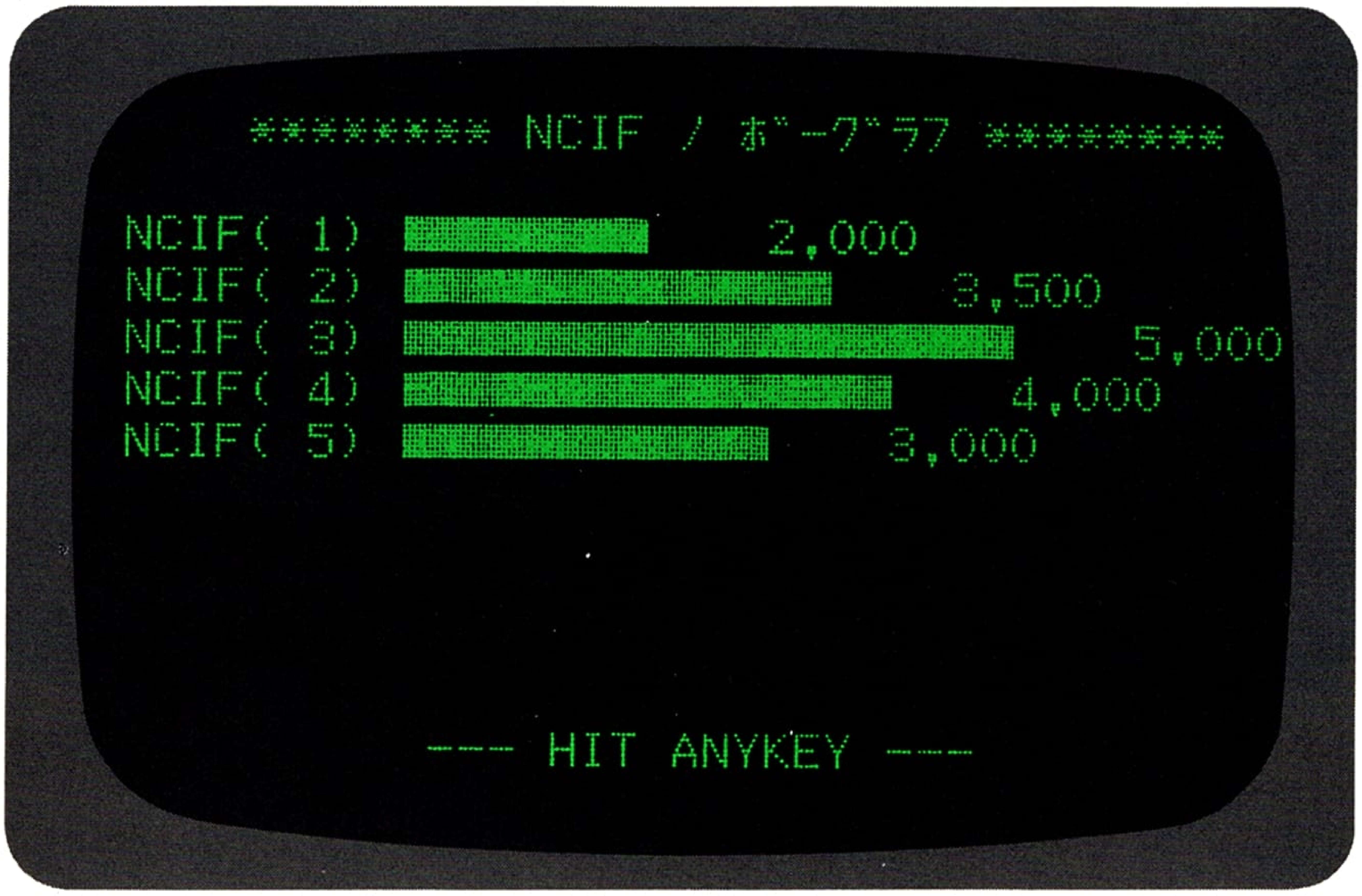
インプットデータの修正

ここで、修正したいデータがあれば、1 ⏎ とし、修正したいデータの番号をキーインして、正しいデータをキーインします。

すべてのデータが正しくキーインできたら、0 ⏎ とします。画面に次の棒グラフが表示されます。

棒グラフの表示

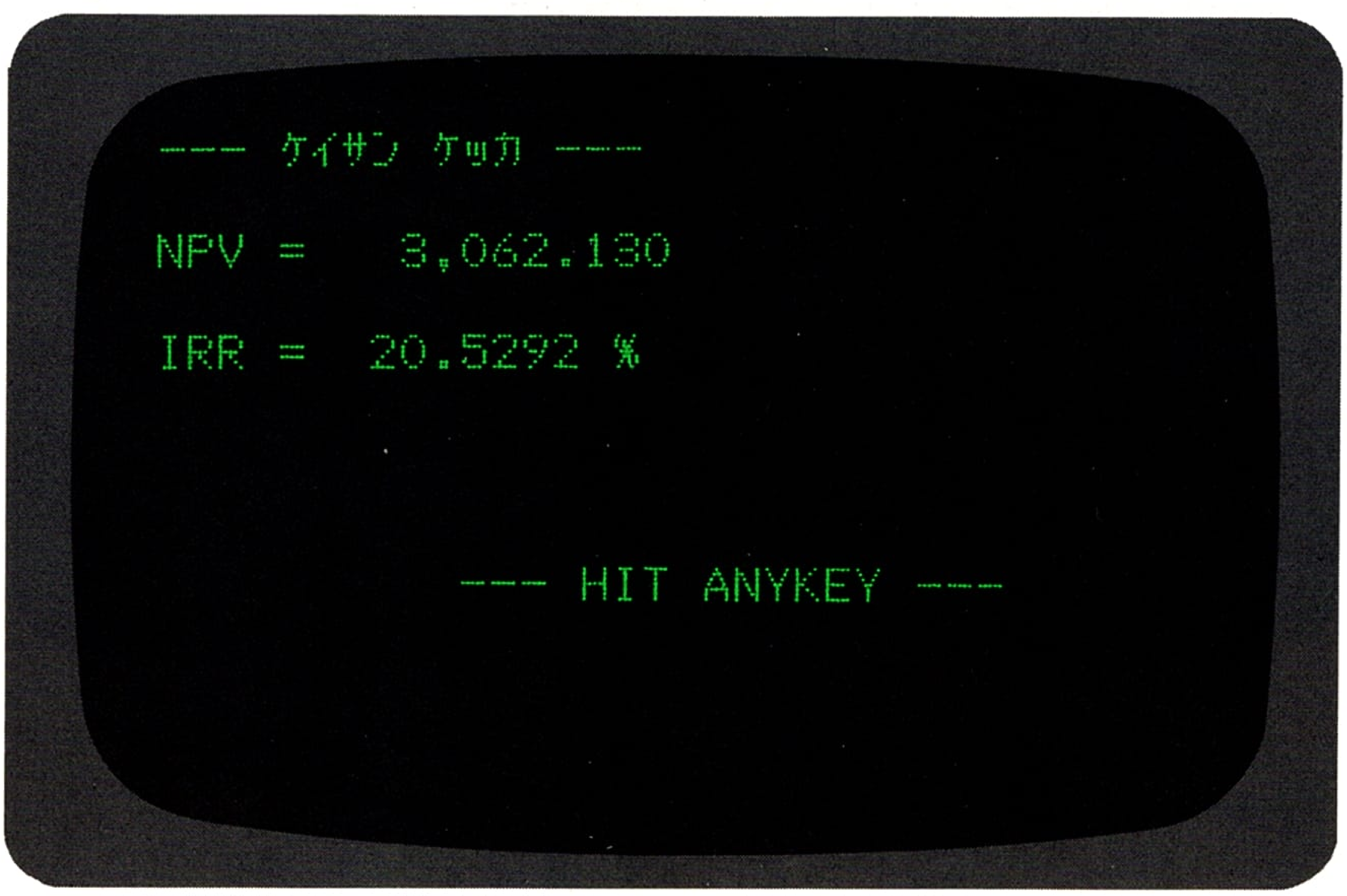
ＮＣＩＦの棒グラフは、次のとおりです。



ここで、どれかキーを押せば、計算結果が画面に表示されます。

計算結果の表示

計算結果は次のとおりです。



どれかキーを押せば、次の画面になります。

オワリ



ここで、どのキーでもいいから押せば、次の画面になり、設備投資計画プログラムの実行が終わります。

設備投資計画プログラムの実行終了



6・8 設備投資計画プログラムのまとめ

設備投資計画プログラム

設備投資計画プログラム

ここでの設備投資計画プログラムは、データをインプットして、正味キャッシュインフロー（ＮＣＩＦ）の棒グラフを作成し、正味現在価値（ＮＰＶ）と内部利益率（ＩＲＲ）を計算するプログラムです。

棒グラフと計算結果をプリンタに出力しよう

ＣＯＰＹ文あるいはＬＰＲＩＮＴ文やＬＰＲＩＮＴ　ＵＳＩＮＧ文を使って、棒グラフと計算結果をプリンタに出力してみてください。

内部利益率（ＩＲＲ）の計算プロセス

内部利益率(ＩＲＲ)の計算プロセス

内部利益率（ＩＲＲ）の計算プロセス（計算過程）をみるためには、ＩＲＲの計算プログラムにおいて、利益率ＲＲと正味現在価値ＮＰＶの途中経過を出力するようにすればよいのです。

ＩＲＲの計算プログラムを次のように修正してみましょう。

```
3200 REM *** IRR ノ ケイサン ***
3210 RR=0
3220 TPV=0
3230 FOR T=1 TO N
3240 TPV=TPV+NCIF(T)/(1+RR/100)^T
3250 NEXT T
3260 NPV=TPV-CI
3263 PRINT " RR =  ";
3264 PRINT USING "##";RR;
3265 PRINT "%";SPC(5);"NPV = ";
3266 PRINT USING "###,###.###";NPV
3270 IF NPV=0 GOTO 3430
3280 IF NPV<0 GOTO 3310
3290 RR=RR+10
3300 GOTO 3220
3310 RR=RR-1
3320 TPV=0
3330 FOR T=1 TO N
3340 TPV=TPV+NCIF(T)/(1+RR/100)^T
3350 NEXT T
3360 NPV=TPV-CI
3363 PRINT " RR =  ";
3364 PRINT USING "##";RR;
```

ＲＲとＮＰＶの途中経過を画面に表示する（3263～3266）

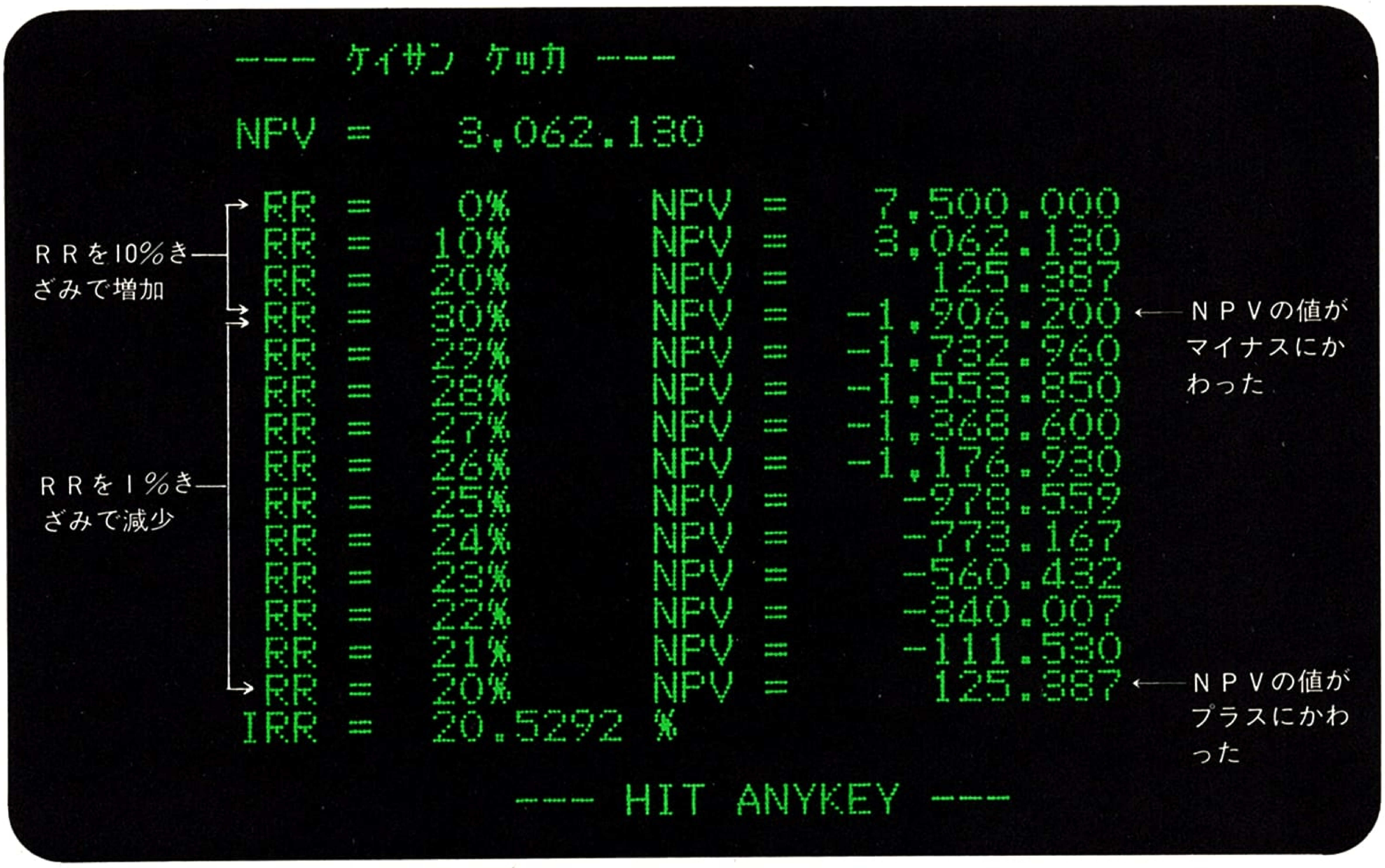
ＲＲの途中経過を画面に表示する（3363～3364）

```
3365 PRINT "%";SPC(5);"NPV = ";
3366 PRINT USING "###,###.###";NPV
3370 IF NPV=0 GOTO 3430
3380 IF NPV>0 GOTO 3410
3390 NPV1=NPV
3400 GOTO 3310
3410 IRR=RR+NPV/(NPV-NPV1)
3420 GOTO 3440
3430 IRR=RR
3440 PRINT "IRR = ";IRR;"%"
3450 GOSUB 5000
```

NPVの途中経過を画面に表示する（3365〜3366）

修正がすんだら、設備投資計画プログラムを実行しましょう。

計算結果は、次のようになります。



この計算結果から、内部利益率（IRR）の計算プロセスがわかります。

まず、0%から10%きざみで利益率RRを増加させ、そのつど正味現在価値NPVを計算します。そして、NPVがプラスの値からマイナスの値にかわったら、こんどはRRを1%きざみで減少させ、そのつどNPVを計算します。

次に、NPVがマイナスの値からプラスの値にかわる直前のRRと、NPVがプラスの値にかわった直後のRRとから、補間法を使って、内部利益率IRRを計算します。なお、NPVがゼロになれば、その

ときのＲＲが求めるＩＲＲです。

補間法によるＩＲＲの計算は、次のとおりです。

```
       RR=21%……NPV=-111.530 ─────────┐
          ∫           ∫                      │
    ┌─IRR=?%……NPV=   0   ─┐              ├─ 125.387-(-111.530)
    │     ∫           ∫       ├─ 125.387   │
    └─ RR=20%……NPV=125.387 ─┘ ───────────┘
```

補間法：$ＩＲＲ=20\%+\dfrac{125.387}{125.387-(-111.530)}=20.5292\%$

なお、ＩＲＲの計算プログラムでは、−111.530はＮＰＶ１に代入されています。

設備投資計画プログラムのセーブ

設備投資計画プログラムが完成したら、カセットテープあるいはディスクにセーブ（ＳＡＶＥ）しておきましょう

より良い設備投資計画プログラムにするには

ここでの設備投資計画プログラムでは、毎年度の正味キャッシュインフロー（ＮＣＩＦ）をデータとしてインプットしました。しかし、このＮＣＩＦをどのようにして計算するかが、実は重要な問題なのです。ここでの設備投資計画プログラムにＮＣＩＦの計算プログラムを織り込めば、より良い設備投資計画プログラムになります。

また、設備投資計画にあたって、正味現在価値（ＮＰＶ）や内部利益率（ＩＲＲ）のほかに、回収期間(Payback Period, PP)などが計算されます。設備投資計画プログラムとしては、ＮＰＶやＩＲＲ以外のものも計算できるようにしておくとよいでしょう。

正味キャッシュインフロー（ＮＣＩＦ）の計算プロセスや回収期間（ＰＰ）の計算方法については、137ページに示した参考書を参照してください。

より良い設備投資計画プログラムを作ってみてください。期待しています。

第7章

利益計画プログラム

7・1　利益計画プログラムのフローチャート

利益計画プログラムとは

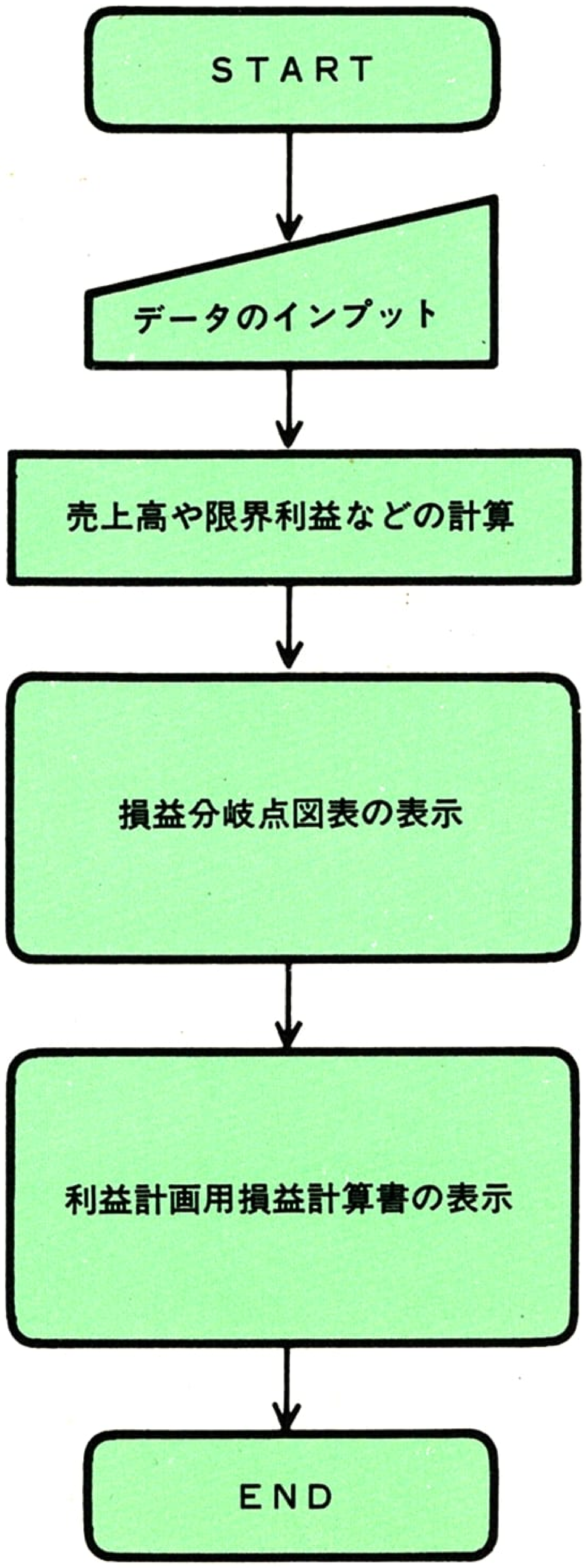
利益計画プログラム

ここでいう利益計画プログラムとは、販売単価、販売数量、単位あたり変動費、固定費のデータにもとづいて、売上高や限界利益などを計算し、損益分岐点図表と利益計画用損益計算書を作成するプログラムをいいます。

利益計画プログラムのフローチャート

利益計画プログラムのフローチャート

利益計画プログラムのフローチャートは、次のとおりです。



それでは、このフローチャートにしたがって、利益計画プログラムの全体の流れを見てみましょう。

まず、利益計画プログラムをスタートさせます。

そして、データをインプットします。ここでは、インプット項目は次の4項目です。

❶ 販売単価………………P(Price)
❷ 販売数量………………SQ(Sales Quantity)
❸ 単位あたり変動費……@VC(Unit Variable Costs)
❹ 固定費…………………FC(Fixed Costs)

続いて、次のものを計算します。

❶ 売上高……………………S(Sales)
❷ 変動費……………………VC(Variable Costs)
❸ 限界利益…………………MP(Marginal Profit)
❹ 営業利益…………………OP(Operating Profit)
❺ 損益分岐点販売数量……BEPSQ(Break-Even Point Sales Quantity)
❻ 損益分岐点売上高………BEPS(Break-Even Point Sales)

これらの計算式は、次のとおりです。

$$S = P \times SQ$$
$$VC = @VC \times SQ$$
$$MP = S - VC$$
$$OP = MP - FC$$
$$BEPSQ = \frac{FC}{P - @VC}$$
$$BEPS = P \times BEPSQ$$

そして、損益分岐点図表を作成し、画面に表示します。

また、利益計画用損益計算書を作成し、画面に表示します。

なお、損益分岐点図表や利益計画用損益計算書など利益計画については、次のものを参照してください。

岡本清著『原価計算(三訂版)』、国元書房、昭和55年、491－512頁。

それでは、このフローチャートにしたがって、利益計画プログラムを作成してみましょう。

7・2 スタート部分のプログラム

スタート部分のプログラムのリスト

利益計画プログラムのスタート部分のプログラムのリストは、次のとおりです。

```
10 REM *********** リエキケイカク ***********
20 REM ***** START *****
30 WIDTH 40,25                        ← 画面を40桁25行にする
40 CONSOLE 0,25,0,1
50 COLOR 4
60 A27$=STRING$(27,"*")
70 A8$=STRING$(8,"*")
80 FOR I=1 TO 5                       ┐
90 PRINT                              │
100 NEXT I                            │
110 FOR I=1 TO 5                      │
120 PRINT SPC(4);A27$                 │
130 NEXT I                            │
140 FOR I=1 TO 2                      │
150 PRINT SPC(4);A8$;SPC(11);A8$      ├ 初期画面の表示
160 NEXT I                            │
170 PRINT SPC(4);A8$;"  リエキケイカク  ";A8$
180 FOR I=1 TO 2                      │
190 PRINT SPC(4);A8$;SPC(11);A8$      │
200 NEXT I                            │
210 FOR I=1 TO 5                      │
220 PRINT SPC(4);A27$                 │
230 NEXT I                            ┘
240 FOR I=1 TO 1000
250 NEXT I
```

このスタート部分のプログラムは、第6章の設備投資計画プログラムのスタート部分のプログラムとほとんど同じです。そこで、利益計画プログラムをキーインするにあたって、まず設備投資計画プログラムをロード（LOAD）し、それを修正すればかなり簡単にキーインできます。

7・3　データインプットのプログラム

データインプットのプログラム

データインプットのプログラムも、設備投資計画プログラムにおけるデータインプットのプログラムによくにています。

データのインプット

データインプットのプログラムは、次のとおりです。

```
1000 REM ***** DATA / INPUT *****
1010 CLS
1020 COLOR 5
1030 M3$=STRING$(3,"-")
1040 PRINT M3$;" data ｦ input ｼﾃｸﾀﾞｻｲ ";M3$
1050 PRINT
1060 DIM N$(4),D(4)
1070 N$(1)="(1) ﾊﾝﾊﾞｲﾀﾝｶ        = "
1080 N$(2)="(2) ﾊﾝﾊﾞｲｽｳﾘｮｳ      = "
1090 N$(3)="(3) ﾀﾝｲｱﾀﾘ ﾍﾝﾄﾞｳﾋ = "
1100 N$(4)="(4) ｺﾃｲﾋ           = "
1110 FOR I=1 TO 4
1120 PRINT N$(I);
1130 INPUT D(I)
1140 NEXT I
```

データのインプット（1110〜1140）

インプットデータは、販売単価、販売数量、単位あたり変動費、固定費の４項目です。

ここで、変動費とは、材料費などのように、製品の生産販売数量に比例して、増加していくような原価です。　**変動費**

それに対して、固定費とは、減価償却費や本社経費のように、製品の生産販売数量とは関係なしに決まるような原価です。　**固定費**

行番号1110〜1140のＦＯＲ〜ＮＥＸＴ文で、これらのデータをインプットします。

インプットデータの画面表示

インプットデータを見やすくするためのプログラムは、次のとおりです。

```
1200 REM *** input data ***
1210 CLS
1220 A3$=STRING$(3,"*")
1230 PRINT A3$;" input data ";A3$
1240 PRINT
1250 FOR I=1 TO 4
1260 PRINT N$(I);
1270 PRINT USING "###,###";D(I)
1280 NEXT I
1290 PRINT
```

- 1250〜1280：インプットデータの画面表示

行番号1250〜1280のＦＯＲ〜ＮＥＸＴ文で、インプットデータを画面に見やすく表示します。

インプットデータの修正

インプットデータを修正するためのプログラムは、次のとおりです。

```
1500 REM *** data ﾉ ｼｭｳｾｲ ***
1510 PRINT " data ｦ ｼｭｳｾｲ ｼﾏｽｶ ?"
1520 PRINT "                    1 ";M3$;" ｽﾙ"
1530 PRINT "                    0 ";M3$;" ｼﾅｲ"
1540 PRINT "                         ";
1550 INPUT CD
1560 IF CD=1 GOTO 1590
1570 IF CD=0 GOTO 2000
1580 GOTO 1500
1590 PRINT
1600 PRINT "ﾅﾝﾊﾞﾝ ﾉ data ｦ ｼｭｳｾｲ ｼﾏｽｶ ";
1610 INPUT I
1620 IF I<=0 OR I>=5 GOTO 1590
1630 PRINT N$(I);
1640 INPUT D(I)
1650 GOTO 1200
```

- 1550：データを修正するなら[1] ⏎、データを修正しないなら[0] ⏎
- 1560：→データの修正へ
- 1570：→計算プログラムへ
- 1610：修正するデータの番号をキーイン
- 1630〜1640：正しいデータをキーイン
- 1650：→インプットデータの画面表示へ

データを修正する（ＣＤ＝１）のであれば、行番号1590のＰＲＩＮＴ文へ飛び、修正するデータの番号をキーインし、正しいデータをキーインします。そして、行番号1650のＧＯＴＯ文により、インプットデータの画面表示へもどります。

すべてのデータが正しくキーインできていれば、行番号1570のＩＦ…ＧＯＴＯ文により、次の計算プログラムへ飛びます。

7・4　計算プログラム

計算プログラムは、次のとおりです。

```
2000 REM ***** ケイサン *****
2010 CLS
2020 COLOR 6
2030 AP=D(1)
2040 SQ=D(2)
2050 AVC=D(3)
2060 FC=D(4)
2070 S=AP*SQ              ←―― 売上高の計算
2080 VC=AVC*SQ            ←―― 変動費の計算
2090 MP=S-VC              ←―― 限界利益の計算
2100 OP=MP-FC             ←―― 営業利益の計算
2110 TC=VC+FC             ←―― 総原価の計算
2120 BEPSQ=FC/(AP-AVC)    ←―― 損益分岐点販売量の計算
2130 BEPS=AP*BEPSQ        ←―― 損益分岐点売上高の計算
```

売上高

売上高Ｓは、販売価格ＡＰに販売数量ＳＱを掛けて計算します。

変動費

変動費ＶＣは、単位あたり（１個あたり）変動費ＡＶＣに販売数量ＳＱを掛けて計算します。

限界利益

限界利益ＭＰとは、売上高Ｓから変動費ＶＣを差し引いた残額で、固定費を回収して営業利益を生み出す原動力となる利益です。

営業利益

この限界利益ＭＰから固定費ＦＣを差し引けば、営業利益ＯＰが計算できます。

総原価

変動費ＶＣと固定費ＦＣの合計が総原価ＴＣ（Total Costs）です。

損益分岐点

売上高Ｓと総原価ＴＣとが等しくなる点、つまり営業利益も営業損失も生じないような点が、損益分岐点ＢＥＰ(Break-Even Point)です。

損益分岐点販売数量
損益分岐点売上高

損益分岐点販売数量ＢＥＰＳＱと損益分岐点売上高Ｓは、次のようにして計算します。

すなわち、損益分岐点ＢＥＰのときは、売上高＝総原価ですから、

ＡＰ＊ＢＥＰＳＱ＝ＡＶＣ＊ＢＥＰＳＱ＋ＦＣとなり、

$$BEPSQ = \frac{FC}{AP - AVC}$$

となります。また、売上高＝販売価格＊販売数量ですから、

$$BEPS = AP * BEPSQ$$

となります。

7・5　損益分岐点図表プログラム

テキスト画面とグラフィック画面

ディスプレイの画面を見てください。あなたの見ている画面は、実は、２枚の同じ大きさの画面を重ねて見ているのです。

テキスト画面

１枚目の画面には、キーボードから入力した文字やプログラムのリストなどを表示します。この画面をテキスト画面と呼びます。

ドット
グラフィック画面

２枚目の画面には、たくさんの点を打つことができます。この点のことをドットと呼びます。ドットを打って、グラフや図形を表示するのです。この画面のことをグラフィック画面と呼びます。

グラフィック画面のモード選択

グラフィック画面のモードには、次の３つがあります。

❶　カラーモード（640×200ドット、１ページ）
❷　白黒モード（640×200ドット、３ページ）
❸　高分解能白黒モード（640×400ドット、１ページ）

ＳＣＲＥＥＮ文

これら３つのモードの切りかえには、次のようなＳＣＲＥＥＮ文を使います。

❶のカラーモード――→　ＳＣＲＥＥＮ △ ０ ⏎
❷の白黒モード――→　ＳＣＲＥＥＮ △ １ ⏎
❸の高分解能白黒モード――→　ＳＣＲＥＥＮ △ ２ ⏎

カラーモード

ここでは、❶のカラーモードを使うことにします。次のようにキーインしてみましょう。

```
SCREEN △0 ⏎ ←――――― グラフィック画面をカラーモードにする
```

画面クリア

テキスト画面をクリアするには、次の方法があります。

ＣＬＳ
ＣＬＳ △ １
CLR キーを押す

❶　ＣＬＳ ⏎
❷　ＣＬＳ △ １ ⏎
❸　CLR キーを押す

ＣＬＳ △ ２

グラフィック画面をクリアするには、次のようにします。

ＣＬＳ △ ２ ⏎

ＣＬＳ △ ３

テキスト画面とグラフィック画面の両方をクリアするには、次のようにします。

ＣＬＳ △ ３ ⏎

ここでは、テキスト画面とグラフィック画面の両方をクリアしてみましょう。

CLS △ 3 ⏎ ← テキスト画面とグラフィック画面の両方をクリア

ディスプレイ画面の各部の色を指定する

まず、次のようにキーインして、テキスト画面をカラーモードにします。

CONSOLE 0,25,0,1 ⏎ ← テキスト画面をカラーモードにする

そして、次のようにキーインします。

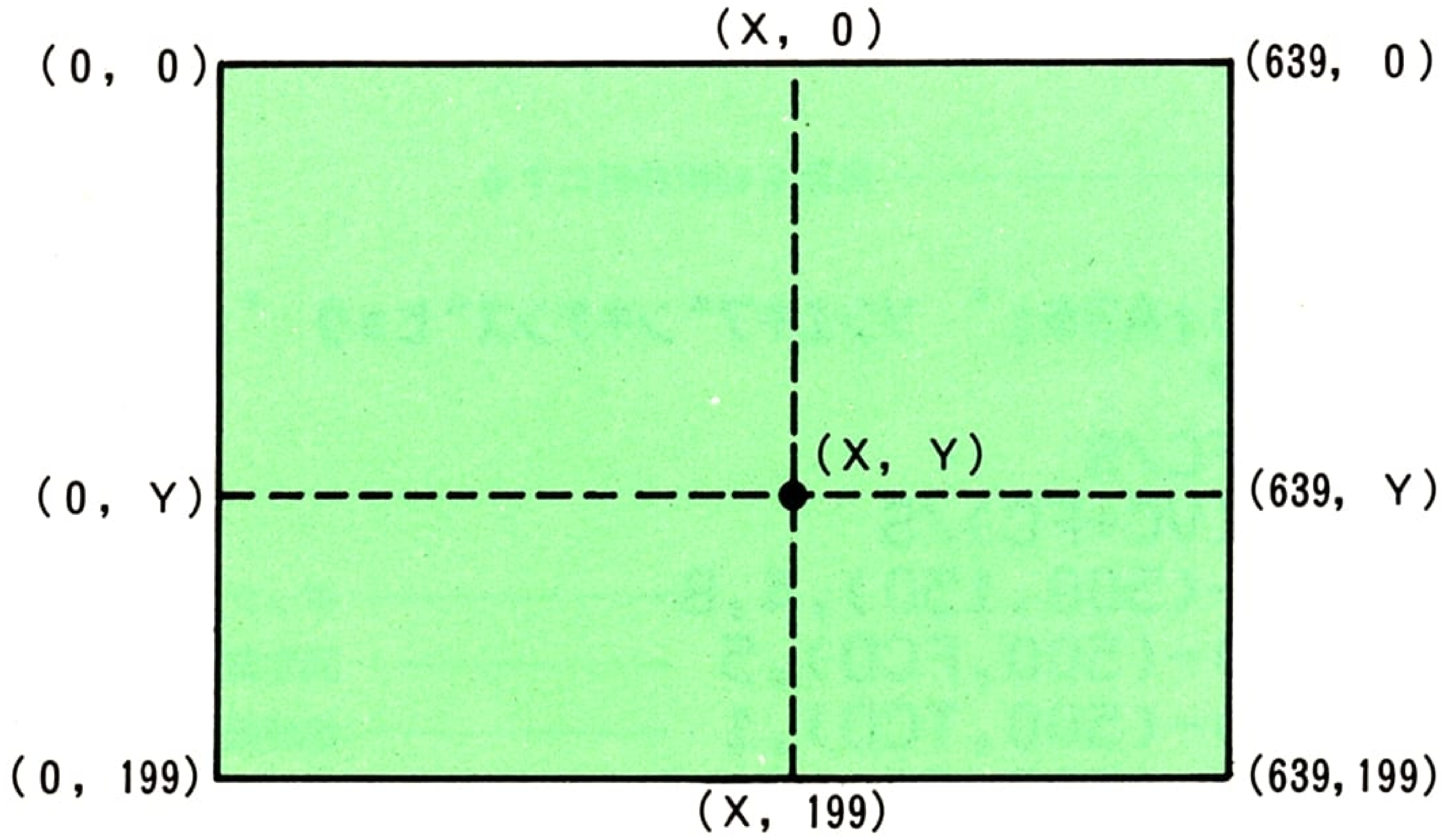
COLOR △7,0,0,7 ⏎ ← ディスプレイ画面の各部の色を指定

COLOR文

このCOLOR文には、4つのパラメータがあります。最初のパラメータ7は、テキスト画面の文字の色を白に指定します。2番目のパラメータ0は、バックグラウンドカラー(グラフィック画面の地の色)を黒に指定します。3番目のパラメータ0は、ボーダーカラー（ディスプレイ画面の中ではあるが、BASICによって使えないエリアの色）を黒に指定します。4番目のパラメータは、フォアグラウンドカラー（グラフィック画面のドットの色）を白に指定します。

グラフィック画面のドットの位置

グラフィック画面は、SCREEN △ 0によって、カラーモードになっています。したがって、グラフィック画面は640×200ドットであり、横に640個、縦に200個、合計640×200＝128,000個のドットを打つことができます。このときドットの位置は、画面の左上隅が（0，0）、右下隅が（639,199）となります。



グラフィック画面に線を引く

LINE文

次のLINE文を使えば、グラフィック画面において、ドット(X_1, Y_1)とドット(X_2, Y_2)を結ぶ直線を引くことができます。

```
LINE (X1, Y1)-LINE(X2, Y2), 色番号 ← グラフィック画面に線を引く
```

次のようにキーインしてみましょう。

```
LINE(100,150)-(500,20),6⏎
```

黄色の直線が引けましたね。

グラフィック画面にボックス（箱）を描く

LINE文

次のLINE文を使えば、グラフィック画面において、ドット(X_1, Y_1)とドット(X_2, Y_2)を対角とする四角形のボックス（箱）を描きます。なお、BはBoxの頭文字Bです。

```
LINE (X1, Y1) -LINE(X2, Y2), 色番号 ,B ← グラフィック画面にボックス(箱)を描く
```

次のようにキーインしてみましょう。

```
LINE(100,20)-(500,150),4,B ⏎
```

緑色のボックス（箱）が描けましたね。

それでは、以上の知識にもとづいて、損益分岐点図表を作成してみましょう。

損益分岐点図表プログラムのリスト

損益分岐点図表プログラム

損益分岐点図表プログラムのリストは、次のとおりです。

```
3000 REM ***** ｿﾝｴｷﾌﾞﾝｷﾃﾝｽﾞﾋｮｳ *****
3010 SCREEN 0
3020 CLS 3
3030 WIDTH 80,25                          ← 画面を80桁25行にする
3040 LOCATE 10,0
3050 PRINT SPC(15);A3$;" ｿﾝｴｷﾌﾞﾝｷﾃﾝｽﾞﾋｮｳ ";A3$
3060 COLOR 7,0,0,7
3070 FCD=150-130*FC/S
3080 TCD=150-130*(VC+FC)/S
3090 LINE(100,20)-(500,150),4,B←          ボックスを描く
3100 LINE(100,FCD)-(500,FCD),5  ←          固定費線を引く
3110 LINE(100,FCD)-(500,TCD),1  ←          総原価線を引く
3120 LINE(100,150)-(500,20),6   ←          売上高線を引く
3130 BEPXD=100+400*BEPS/S
```

```
3140 BEPYD=150-130*BEPS/S
3150 LINE(BEPXD,BEPYD)-(BEPXD,150),2,,&HAAAA
3160 LOCATE 0,2
3170 PRINT "S   = ";
3180 PRINT USING "###,###";S
3190 LOCATE 0,15
3200 PRINT "FC = ";
3210 PRINT USING "###,###";FC
3220 LOCATE 31,20
3230 PRINT "BEPS  = ";
3240 PRINT USING "###,###";BEPS
3250 LOCATE 31,21
3260 PRINT "BEPSQ = ";
3270 PRINT USING "###,###";BEPSQ
3280 LOCATE 65,2
3290 PRINT "S   = ";
3300 PRINT USING "###,###";S
3310 LOCATE 65,5
3320 PRINT "TC = ";
3330 PRINT USING "###,###";TC
3340 LOCATE 65,6
3350 PRINT "VC = ";
3360 PRINT USING "###,###";VC
3370 LOCATE 65,15
3380 PRINT "FC = ";
3390 PRINT USING "###,###";FC
3400 LOCATE 65,20
3410 PRINT "S   = ";
3420 PRINT USING "###,###";S
3430 LOCATE 65,21
3440 PRINT "SQ = ";
3450 PRINT USING "###,###";SQ
3460 GOSUB 6000
```

損益分岐点から横軸に垂線を点線でおろす（3150）

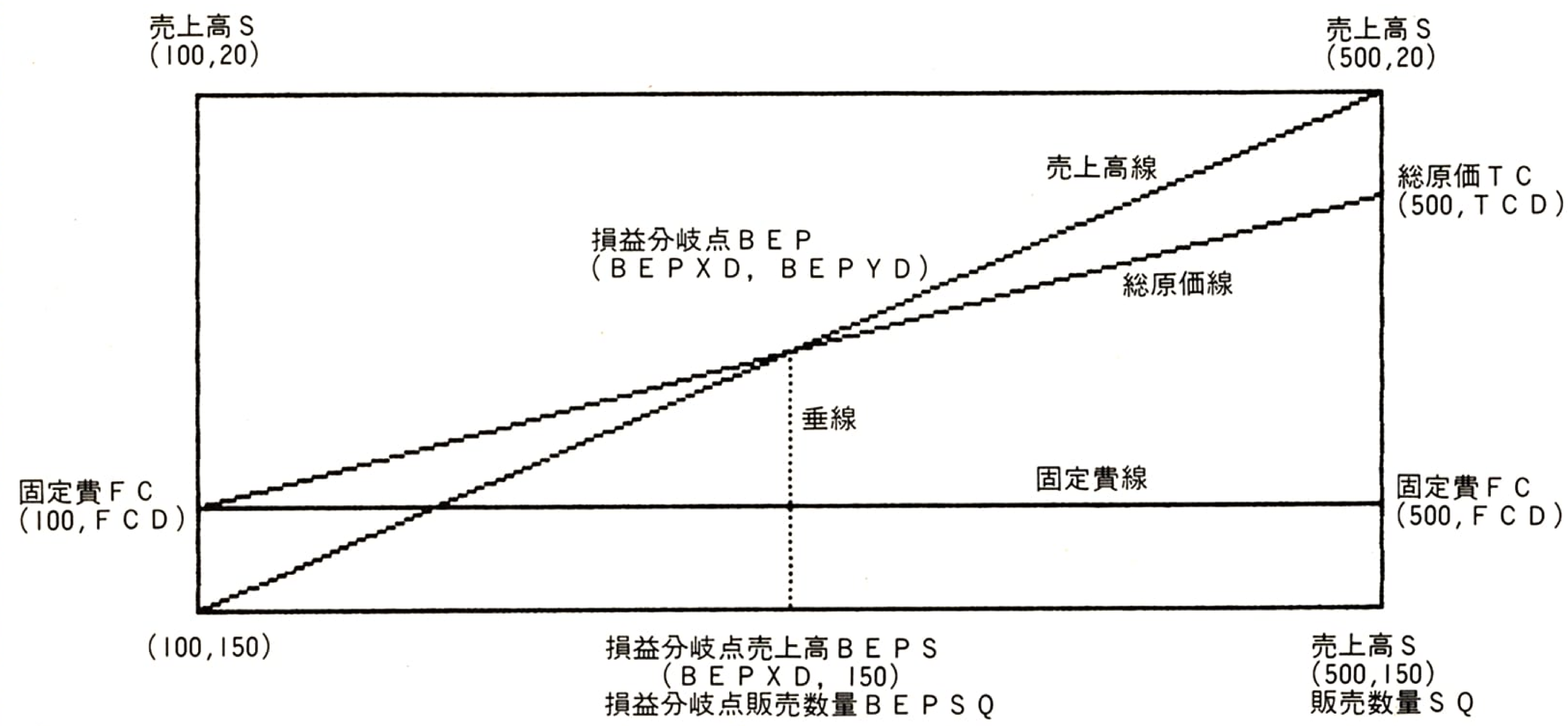
ＧＯＳＵＢ文（3460）

損益分岐点図表におけるドットの位置は、次ページのとおりです。

行番号3070の代入文では、固定費ＦＣのＹ座標ＦＣＤを計算します。まず、固定費のＹ座標を売上高に対する固定費の比率ＦＣ／Ｓであらわし、縦軸の範囲130（150－20＝130）ドット倍します。そして、それを150から差し引いてＹ座標とします。

同じようにして、行番号3080の代入文では、総原価ＴＣのＹ座標ＴＣＤを計算します。

行番号3090のＬＩＮＥ文で、損益分岐点図表の緑色ボックス（箱）を描きます。続いて、行番号3100のＬＩＮＥ文で青色の固定費線を引



き、行番号3200のＬＩＮＥ文で水色の総原価線を引き、行番号3120のＬＩＮＥ文で黄色の売上高線を引きます。売上高線と総原価線の交点が損益分岐点ＢＥＰです。

行番号3130の代入文で、損益分岐点ＢＥＰのＸ座標ＢＥＰＸＤを計算します。まず、損益分岐点のＸ座標を売上高に対する損益分岐点売上高の比率ＢＥＰＳ／Ｓであらわし、横軸の範囲400（500－100＝400)ドット倍します。そして、それに100を加えてＸ座標とします。つづいて、行番号3140の代入文で損益分岐点のＹ座標ＢＥＰＹＤを計算します。

行番号3150のＬＩＮＥ文で、損益分岐点ＢＥＰから横軸に垂線を紫色の点線でおろします。

グラフィック画面に点線を引くには、次のＬＩＮＥ文を使います。

```
LINE(X1,Y1)-(X2,Y2),色番号,,&HAAAA   ←―― グラフィック画面に点線を引く
```

ＬＯＣＡＴＥ文
ＰＲＩＮＴ　ＵＳＩＮＧ文

次に、ＬＯＣＡＴＥ文を使って、テキスト画面におけるカーソルの位置を指定します。そして、ＰＲＩＮＴ　ＵＳＩＮＧ文を使って、データや計算結果をテキスト画面に出力します。

7・6　利益計画用損益計算書プログラム

利益計画用損益計算書プログラムのリスト

利益計画用損益計算書プログラムのリストは、次のとおりです。　**利益計画用損益計算書プログラム**

```
4000 REM ***** ｿﾝｴｷｹｲｻﾝｼｮ *****
4010 CLS
4020 WIDTH 40,25
4030 COLOR 4
4040 PRINT SPC(8);A3$;" ｿﾝｴｷｹｲｻﾝｼｮ ";A3$
4050 PRINT
4060 PRINT
4070 PRINT "ｳﾘｱｹﾞﾀﾞｶ";
4080 PRINT USING "###,###";AP;
4090 PRINT " * ";
4100 PRINT USING "###,##";SQ;
4110 PRINT " = ";
4120 PRINT USING "###,###";S
4130 PRINT "ﾍﾝﾄﾞｳﾋ   ";
4140 PRINT USING "###,###";AVC;
4150 PRINT " * ";
4160 PRINT USING "###,##";SQ;
4170 PRINT " = ";
4180 PRINT USING "###,###";VC
4190 PRINT SPC(27);"────────"
4200 PRINT "    ｹﾞﾝｶｲﾘｴｷ  ";SPC(15);
4210 PRINT USING "###,###";MP
4220 PRINT "ｺﾃｲﾋ            ";SPC(15);
4230 PRINT USING "###,###";FC
4240 PRINT SPC(27);"────────"
4250 PRINT "    ｴｲｷﾞｮｳﾘｴｷ";SPC(15);
4260 PRINT USING "###,###";OP
4270 PRINT SPC(27);"════════"
4280 GOSUB 6000
```

（4020：画面を40桁25行にする）
（4190・4240：1本線の横線を引く）
（4270：2本線の横線を引く）
（4280：GOSUB文）

利益計画用損益計算書は、売上高から変動費を差し引いて限界利益を表示し、限界利益から固定費を差し引いて営業利益を示す損益計算書です。なお、1本線の横線を引くには、(GRAPH)キーを押しながらテンキーの[*]（アスタリスク）キーを押し続ければよいのです。また2本線の横線を引くには、(GRAPH)キーを押しながらテンキーの[+]（プラス）キーを押し続ければよいのです。　**利益計画用損益計算書**

7・7 エンド部分とサブルーチンのプログラム

エンド部分のプログラムのリスト

利益計画プログラムのエンド部分のプログラムのリストは、次のとおりです。

```
5000 REM ***** END *****
5010 CLS
5020 COLOR 4
5030 PRINT M3$;" ｵﾜﾘ ";M3$
5040 PRINT
5050 PRINT " ｵﾂｶﾚｻﾏﾃﾞｼﾀ "
5060 PRINT
5070 PRINT " ｵﾜﾘﾃﾞｽ "
5080 GOSUB 6000 ←―――――――――― GOSUB文
5090 CLS
5100 WIDTH 80,20
5110 CONSOLE 0,20,1,0
5120 END
```

サブルーチンプログラムのリスト

サブルーチンプログラムのリストは、次のとおりです。

```
6000 REM ***** SUB (HIT ANYKEY) *****
6010 COLOR 1
6020 LOCATE 29,23
6030 PRINT M3$;" HIT ANYKEY ";M3$
6040 Z$=INKEY$
6050 IF Z$="" GOTO 6020
6060 BEEP
6070 RETURN
```

利益計画プログラムのリスト

ＬＩＳＴコマンドを使って、利益計画プログラムを画面に表示してみましょう。

また、ＬＬＩＳＴコマンドを使って、利益計画プログラムのリストをプリンタに出力してみてください。

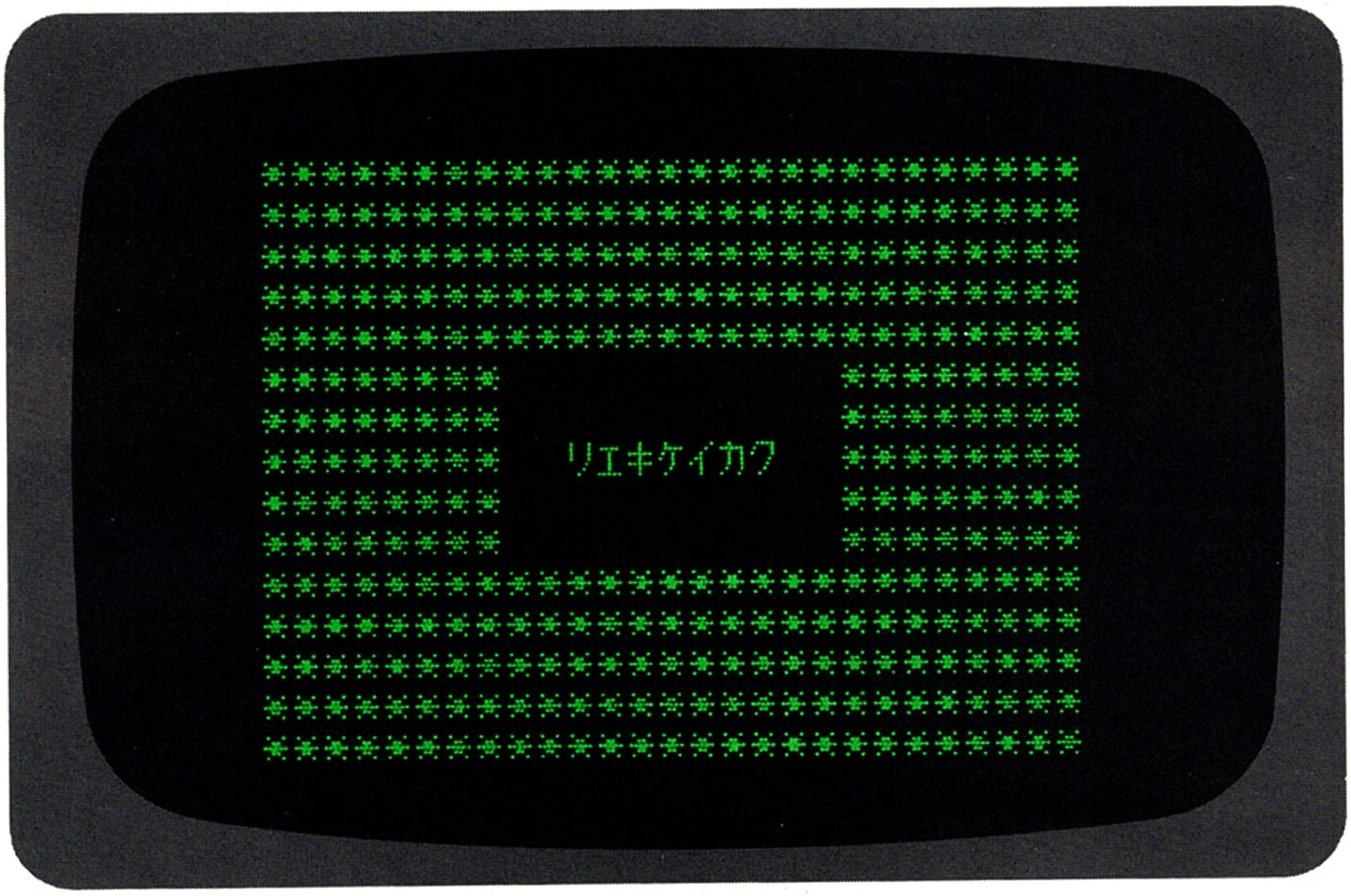
7·8　利益計画プログラムの実行

RUN

ＲＵＮコマンドを使って、利益計画プログラムを実行してみましょう。

初期画面

利益計画プログラムを実行すると、まず、ディスプレイに次のような初期画面が表示されます。



データのインプット

次に、画面のメッセージにしたがって、データをキーインします。いま、次の画面のようにデータをキーインしたとします。すなわち、

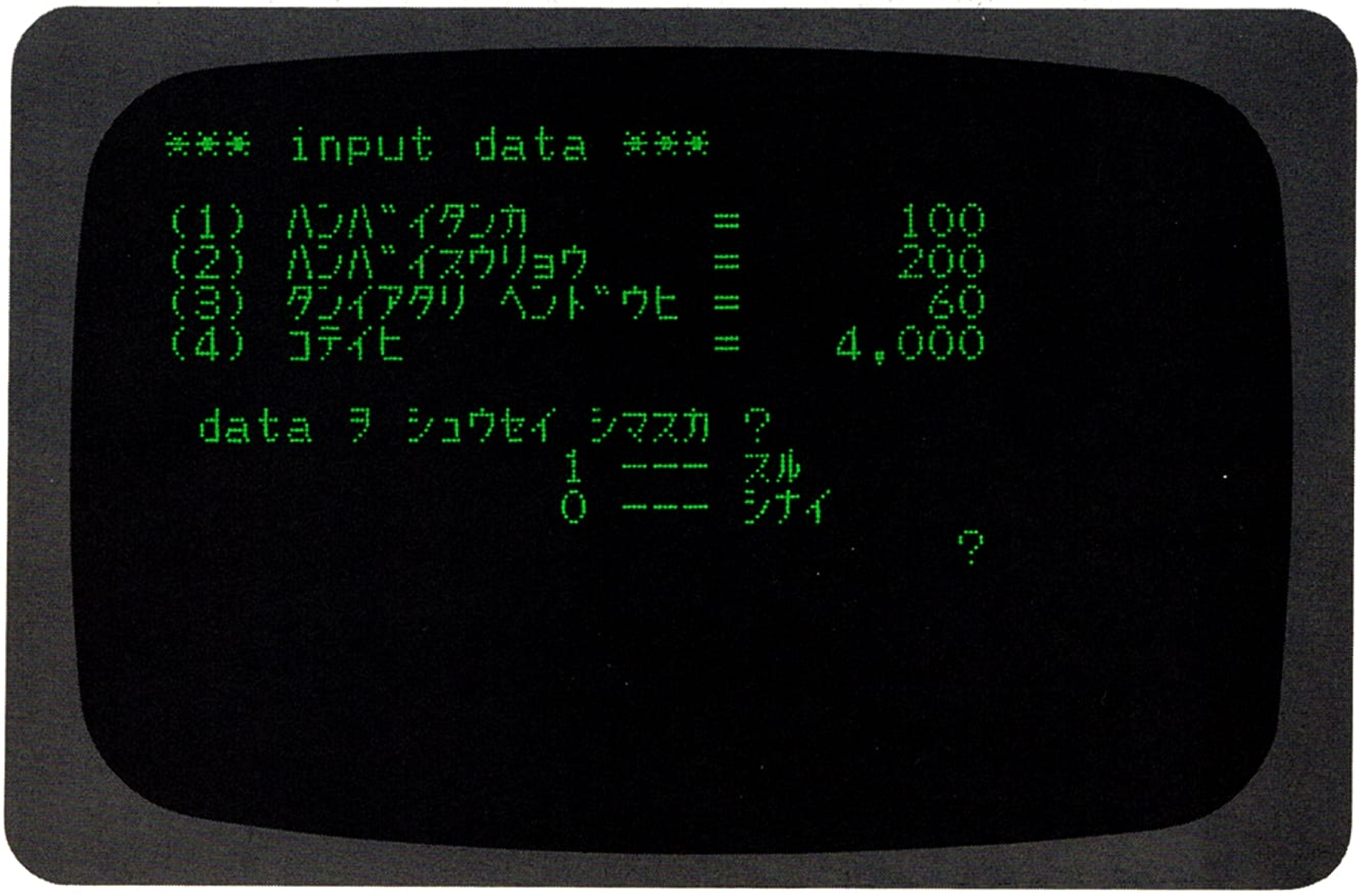
(1)　販売単価＝100円
(2)　販売数量＝200万個
(3)　単位あたり変動費＝60円
(4)　固定費＝4,000万円

です。



インプットデータの見やすい画面表示


ここで、⏎キーを押すと、インプットデータが画面に見やすく表示されます。



インプットデータの修正

修正したいデータがあれば、1 ⏎とし、修正したいデータの番号をキーインして、正しいデータをキーインします。すべてのデータが正しくキーインできたら、0 ⏎とします。

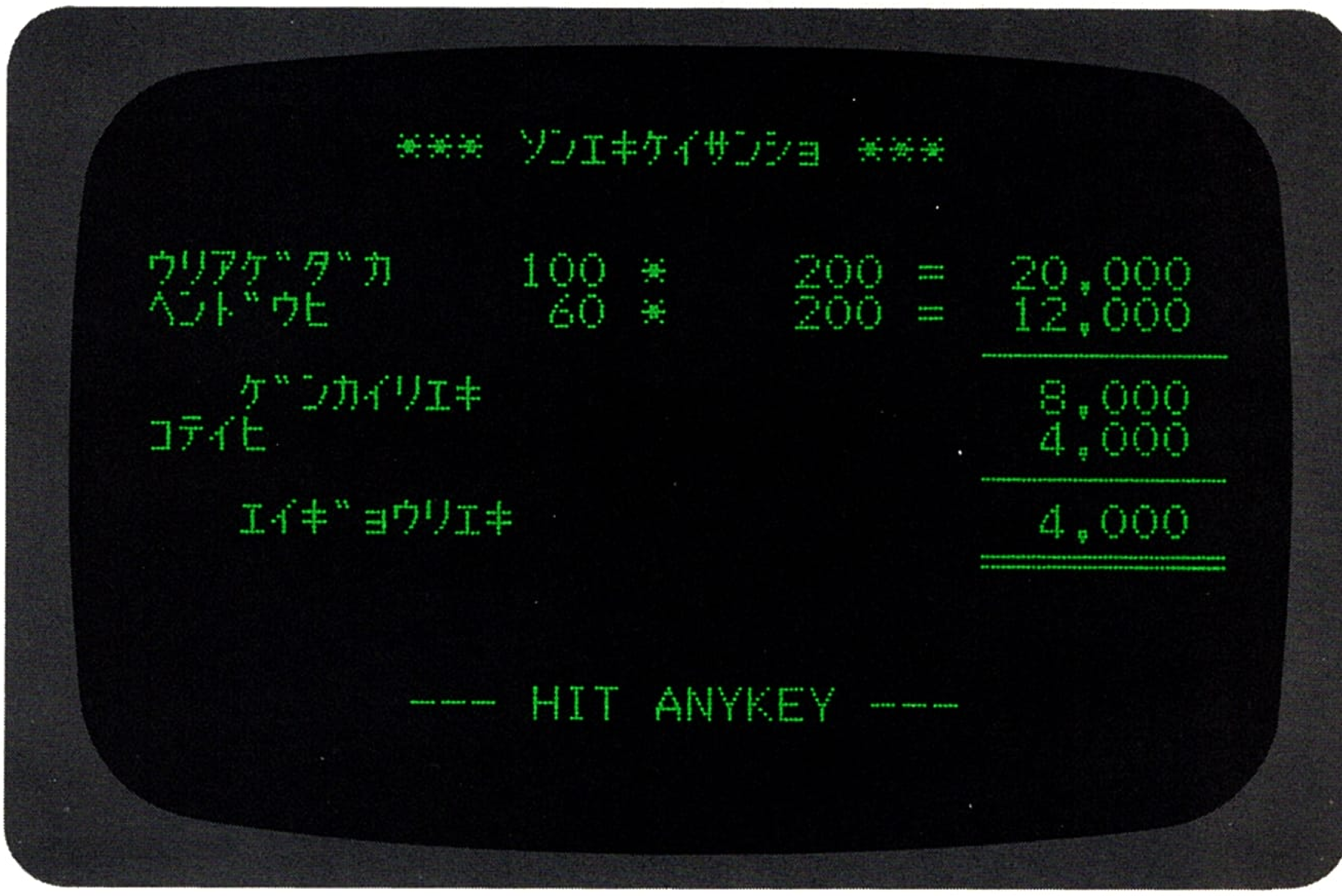
損益分岐点図表の表示


*** ｿﾞﾝｴｷﾌﾞﾝｷﾃﾞﾝｽﾞﾋｮｳ ***
S = 20,000
TC = 16,000
VC = 12,000
FC = 4,000
S = 20,000
SQ = 200
S = 20,000
FC = 4,000
BEPS = 10,000
BEPSQ = 100
--- HIT ANYKEY ---

ここで、どれかキーを押せば、利益計画用の損益計算書が画面に表示されます。

利益計画用の損益計算書の表示

利益計画用の損益計算書は、次のとおりです。



損益計算書は、単位を明確にすれば、次のようになります。

<u>損益計算書</u>

売上高	@100円×200万個＝	20,000万円
変動費	@ 60円×200万個＝	12,000万円
限界利益		8,000万円
固定費		4,000万円
営業利益		4,000万円

単位を表示するよう利益計画用損益計算書を修正してみてください。

なお、注意すべきは、ＰＲＩＮＴ　ＵＳＩＮＧ文を使って、数値を〝＃＃＃，＃＃＃〞のかたちで出力するようにしていることです。必要があれば、小数点以下の値も出力できるように修正してみましょう。また、売上高などの数値の桁数が大きくなることが予想されれば、売上高の変数Ｓを倍精度実数型変数Ｓ＃にするなど、倍精度実数型変数を使ってみましょう。

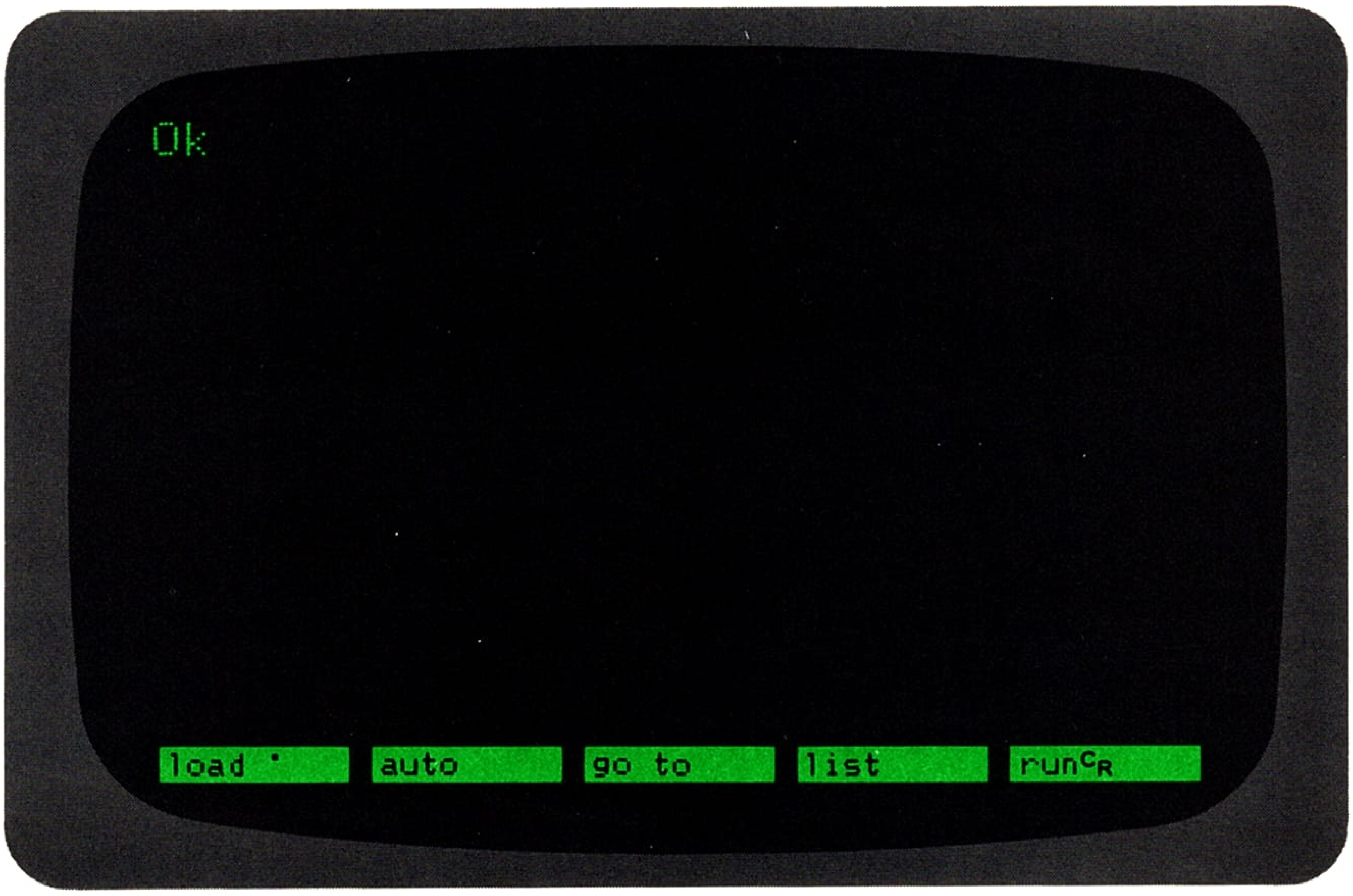
さて、ここでどれかキーを押せば、次の画面になります。

オワリ



ここで、どのキーでもいいから押せば、次の画面になり、利益計画プログラムの実行が終わります。

利益計画プログラムの実行の終了



7・9　利益計画プログラムのまとめ

利益計画プログラム

ここでの利益計画プログラムは、販売単価、販売数量、単位あたり変動費、固定費のデータにもとづいて、売上高や限界利益などを計算し、損益分岐点図表と利益計画用損益計算書を作成するプログラムです。

なお、利益計画といっても、長期や中期の利益計画ではありません。１ヵ月、半年、１年などの短期の利益計画です。

損益分岐点図表と損益計算書をプリンタに出力しよう

ＣＯＰＹ文あるいはＬＰＲＩＮＴ文やＬＰＲＩＮＴ　ＵＳＩＮＧ文を使って、損益分岐点図表と損益計算書をプリンタに出力してみてください。

利益計画プログラムのセーブ

利益計画プログラムが完成したら、セーブ（ＳＡＶＥ）しておきましょう。

より良い利益計画プログラムにするには

損益分岐点売上高と損益分岐点販売数量だけでなく、次のものを計算して表示できるプログラムにすれば、より良い利益計画プログラムになります。

❶　目標営業利益をあげるのに必要な売上高

❷　目標営業利益をあげるのに必要な販売数量

❸　目標売上高営業利益率をあげるのに必要な売上高

❹　目標売上高営業利益率をあげるのに必要な販売数量

❺　$\text{安全余裕率}(\%) = \dfrac{\text{売上高}-\text{損益分岐点売上高}}{\text{売　上　高}} \times 100$

また、変動費や固定費のデータを、変動直接材料費、変動直接労務費、変動販売費、固定製造間接費、固定販売費および一般管理費などに分類して入力するようにすれば、もっとくわしい利益計画用損益計算書を作成することができます。

より良い利益計画プログラムにするようくふうしてみてください。

第8章

予算実績差異分析プログラム

8・1 予算実績差異分析プログラム

予算実績差異分析プログラムとは

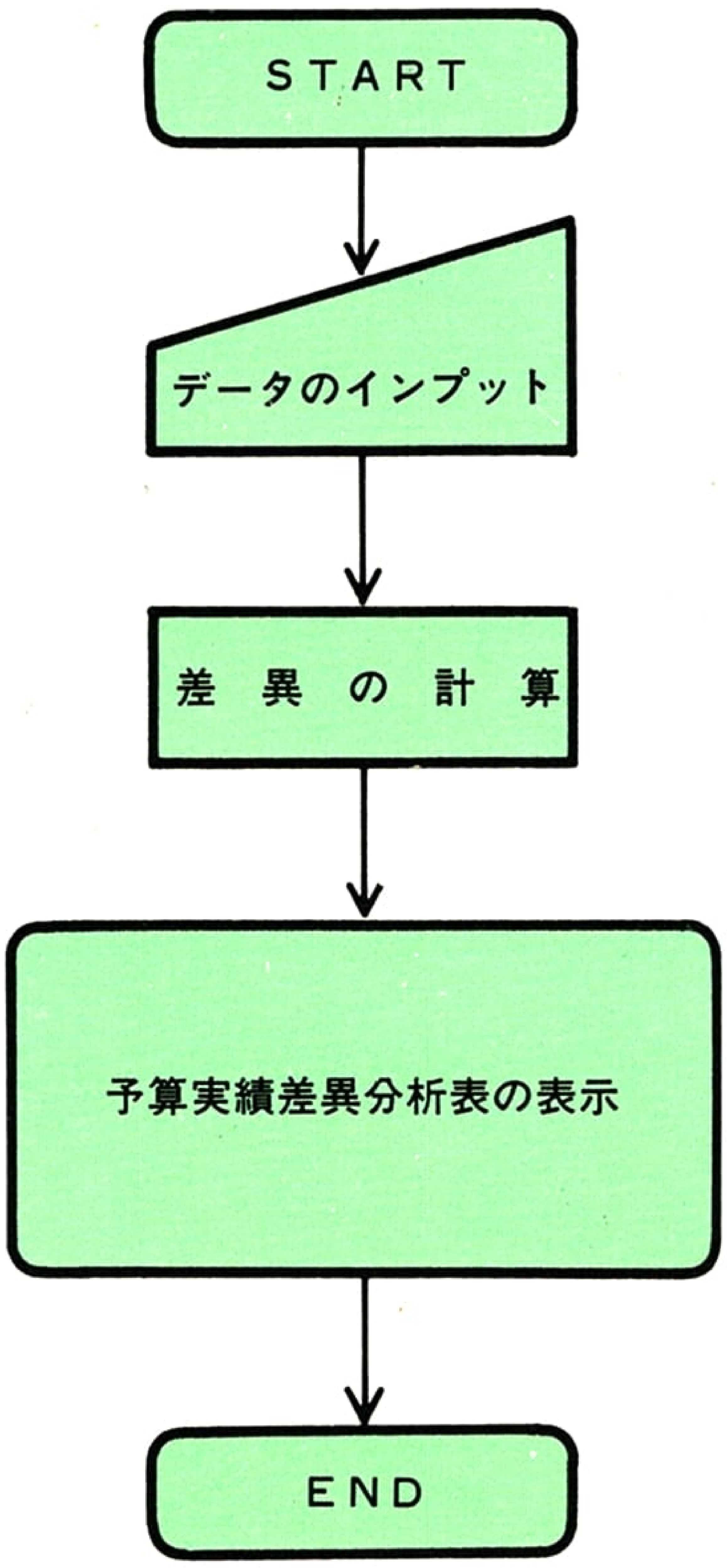
予算実績差異分析プログラム

ここでいう予算実績差異分析プログラムとは、売上高、変動費、限界利益、固定費、営業利益の予算データと実績データにもとづいて、それらの差異を計算し、予算実績差異分析表（タイトルおよび見出しを漢字で出力する）を作成するプログラムをいいます。

予算実績差異分析プログラムのフローチャート

予算実績差異分析プログラムのフローチャート

予算実績差異分析プログラムのフローチャートは、次のとおりです。



それでは、このフローチャートにしたがって、予算実績差異分析プログラムの全体の流れを見てみましょう。

まず、予算実績差異分析プログラムをスタートさせます。

そして、データをインプットします。インプットデータは次の５項目です。

❶ 売上高
❷ 変動費
❸ 限界利益
❹ 固定費
❺ 営業利益

内部ファイル
ＤＡＴＡ文
ＲＥＡＤ文

これらの項目の予算データは、内部ファイルとして、ＤＡＴＡ文を使ってプログラムの中に用意しておきます。そして、ＲＥＡＤ文によって、予算データをＤＡＴＡ文から読み込みます。なお、内部ファイル、ＤＡＴＡ文、ＲＥＡＤ文については、「８・３　データインプットと計算のプログラム」で説明します。

また５項目について、それぞれ、実績データから予算データを差し引いて差異（差額）を計算します。

予算実績差異分析表
漢字

そして、予算実績差異分析表を作成し、画面に表示します。この場合、タイトルや見出しを漢字で出力します。しかし、漢字を出力するためには、ＰＣ-８８０１本体内にＰＣ-８８０１－０１(ＰＣ-８８０１用漢字ＲＯＭボード）が実装されていなければなりません。

予算実績差異分析表

	予 算	実 績	差 異
売上高	5,000	5,300	300
変動費	2,500	2,700	200
限界利益	2,500	2,600	100
固定費	1,300	1,300	0
営業利益	1,200	1,300	100

なお、予算実績差異分析については、次のものを参照してください。

桜井通晴著『経営原価計算論〔増補版〕』、中央経済社、昭和56年、496－509頁。

それでは、このフローチャートにしたがって、予算実績差異分析プログラムを作成してみましょう。

8·2 スタート部分のプログラム

スタート部分のプログラムのリスト

予算実績差異分析プログラムのスタート部分のプログラムのリストは、次のとおりです。

```
10 REM ********** ﾖｻﾝ ｼﾞｯｾｷ ｻｲﾌﾞﾝｾｷ **********
20 REM ***** START *****
30 WIDTH 40,25
40 CONSOLE 0,25,0,1
50 COLOR 4
60 A37$=STRING$(37,"*")
70 A8$=STRING$(8,"*")
80 FOR I=1 TO 5
90 PRINT
100 NEXT I
110 FOR I=1 TO 5
120 PRINT SPC(2);A37$
130 NEXT I
140 FOR I=1 TO 2
150 PRINT SPC(2);A8$;SPC(21);A8$
160 NEXT I
170 PRINT SPC(2);A8$;"  ﾖｻﾝ ｼﾞｯｾｷ ｻｲﾌﾞﾝｾｷ";
    A8$
180 FOR I=1 TO 2
190 PRINT SPC(2);A8$;SPC(21);A8$
200 NEXT I
210 FOR I=1 TO 5
220 PRINT SPC(2);A37$
230 NEXT I
240 FOR I=1 TO 1000
250 NEXT I
```

初期画面の表示（80〜230）

このスタート部分のプログラムは、第7章の利益計画プログラムのスタート部分のプログラムとほとんど同じです。予算実績差異分析プログラムのキーインにあたって、簡単にキーインできるようなくふうをしてください。

8・3　データインプットと計算のプログラム

データインプットのプログラム

データインプットのプログラムも、利益計画プログラムにおけるデータインプットのプログラムによくにています。しかし、予算データについては、ＲＥＡＤ文によって、ＤＡＴＡ文から読み込みます。

データのインプット

データのインプットのプログラムは、次のとおりです。

```
1000 REM ***** DATA ﾉ INPUT *****
1010 CLS
1020 DIM N$(5),B(5),A(5),V(5)
1030 N$(1)="(1) ｳﾘｱｹﾞﾀﾞｶ   = "
1040 N$(2)="(2) ﾍﾝﾄﾞｳﾋ     = "
1050 N$(3)="(3) ｹﾞﾝｶｲﾘｴｷ   = "
1060 N$(4)="(4) ｺﾃｲﾋ       = "
1070 N$(5)="(5) ｴｲｷﾞｮｳﾘｴｷ  = "
1080 DATA 5000,2500,2500,1300,1200  ← 予算データファイル
1090 FOR I=1 TO 5   ┐
1100 READ B(I)      ├ 予算データを読み込む
1110 NEXT I         ┘
1120 COLOR 5
1130 M3$=STRING$(3,"-")
1140 PRINT M3$;" ｼﾞｯｾｷ ﾃﾞｰﾀ ｦ input
     ｼﾃｸﾀﾞｻｲ ";M3$
1150 PRINT
1160 FOR I=1 TO 5   ┐
1170 PRINT N$(I);   ├ 実績データのインプット
1180 INPUT A(I)     │
1190 NEXT I         ┘
```

インプットデータは、売上高、変動費、限界利益、固定費、営業利益の５項目です。これらの項目には、予算データと実績データの２つがあります。

ＤＡＴＡ文

予算データは、内部ファイルとして、次のＤＡＴＡ文を使ってプログラムの中に用意しておきます。

```
DATA △ 定数，定数，……
```

１つのＤＡＴＡ文には、１行(255文字）に入るだけのデータをセットすることができます。また、ＤＡＴＡ文中のデータはコンマ（ ,）によって区切られています。

内部ファイル

行番号1080のＤＡＴＡ文が、内部ファイルとしてプログラムの中に用意されている予算データファイルです。

ＲＥＡＤ文

ＤＡＴＡ文中のデータを読み込むには、次のＲＥＡＤ文を使います。

```
READ △ 変名数
```

ＲＥＡＤ文は、ＤＡＴＡ文中のデータをＲＥＡＤ文中の変数に１対１に対応させて割りあてます。

そこで、行番号1090～1110のＦＯＲ～ＮＥＸＴ文を実行すると、次のようにＤＡＴＡ文中の予算データがＲＥＡＤ文中の変数に代入されます。

Ｂ（１）＝5,000
Ｂ（２）＝2,500
Ｂ（３）＝2,500
Ｂ（４）＝1,300
Ｂ（５）＝1,200

また、行番号1160～1190のＦＯＲ～ＮＥＸＴ文によって、実績データをインプットします。

実績データの画面表示

実績データを見やすくするためのプログラムは、次のとおりです。

```
1300 REM *** ｼﾞｯｾｷ ﾃﾞｰﾀ ***
1310 CLS
1320 A3$=STRING$(3,"*")
1330 PRINT A3$;" ｼﾞｯｾｷ ﾃﾞｰﾀ ";A3$
1340 PRINT
1350 FOR I=1 TO 5          ←┐
1360 PRINT N$(I);            ├── 実績データの画面表示
1370 PRINT USING "###,###";A(I) │
1380 NEXT I                ←┘
1390 PRINT
```

行番号1350～1380のＦＯＲ～ＮＥＸＴ文で、実績データを画面に見

やすく表示します。

インプットデータの修正

インプットデータを修正するためのプログラムは、次のとおりです。

```
1500 REM *** デ゛ータ ノ シュウセイ ***
1510 PRINT " デ゛ータ ヲ シュウセイ シマスカ ?"
1520 PRINT "                    1 ";M3$;" スル"
1530 PRINT "                    0 ";M3$;" シナイ"
1540 PRINT "                                  ";
1550 INPUT CD
1560 IF CD=1 GOTO 1590
1570 IF CD=0 GOTO 2000
1580 GOTO 1500
1590 PRINT
1600 PRINT "ナンハ゛ン ノ デ゛ータ ヲ シュウセイ シマスカ ";
1610 INPUT I
1620 IF I<=0 OR I>=6 GOTO 1590
1630 PRINT N$(I);
1640 INPUT A(I)
1650 GOTO 1300
```

計算プログラム

実績データから予算データを差し引いて差異を計算する計算プログラムは、次のとおりです。

```
2000 REM ***** ケイサン *****
2010 CLS
2020 FOR I=1 TO 5
2030 V(I)=A(I)-B(I)
2040 NEXT I
```

実績データから予算データを差し引いて、差異を計算する

行番号2020～2040のＦＯＲ～ＮＥＸＴ文によって、実績データ Ａ（Ｉ）から予算データＢ（Ｉ）を差し引いて差異を計算し、Ｖ（Ｉ）に代入します。

8·4 予算実績差異分析表プログラム

予算実績差異分析プログラムのリスト

予算実績差異分析プログラム

予算実績差異分析プログラムのリストは、次のとおりです。

```
3000 REM ***** ﾖｻﾝ ｼﾞｯｾｷ ｻｲﾌﾞﾝｾｷ ﾋｮｳ *****
3010 SCREEN 0          ←グラフィック画面をカラーモードにする
3020 CLS 3             ←テキスト画面とグラフィック画面の両方をクリア
3030 DATA 4D3D,3B3B,3C42,4053,3A39,305B,4A2C,
     404F,493D         ←漢字ＪＩＳコードの用意
3040 N=9 : X=159 : Y=0 : S=32
3050 GOSUB 5000        ←漢字タイトルを画面に表示
3060 N=6 : X=223 : Y=31 : S=64
3070 RESTORE 3030
3080 GOSUB 5000        ←漢字見出しを画面に表示
3090 FOR I=1 TO 3
3100 READ KJC$
3110 NEXT I
3120 DATA 4764,3E65,3962
3130 N=3 : X=15 : Y=63 : S=32
3140 GOSUB 5000        ←┐
3150 DATA 4A51,4630,4871 │
3160 N=3 : X=15 : Y=79 : S=32
3170 GOSUB 5000        ←┤
3180 DATA 3842,3326,4D78,3157
3190 N=4 : X=47 : Y=103 : S=32
3200 GOSUB 5000        ←┼─漢字見出しを画面に表示
3210 DATA 3847,446A,4871
3220 N=3 : X=15 : Y=119 : S=32
3230 GOSUB 5000        ←┤
3240 DATA 3144,3648,4D78,3157
3250 N=4 : X=47 : Y=143 : S=32
3260 GOSUB 5000        ←┘
3500 LOCATE 9,2
3510 PRINT "────────────────────"
3520 PRINT : PRINT : PRINT
3530 GOSUB 7000        ←┐ 1本線の横線を引く
3540 I=1                 │
3550 ON I GOTO 3570,3570,3560,3570,3560,3600
3560 GOSUB 7000        ←┘
3570 GOSUB 6000        ←数値を画面に表示
```

```
3580 I=I+1
3590 GOTO 3550
3600 GOSUB 8000  ←── 2本線の横線を引く
3610 GOSUB 9000  ←── HIT ANYKEY
```

漢字を画面に表示するには

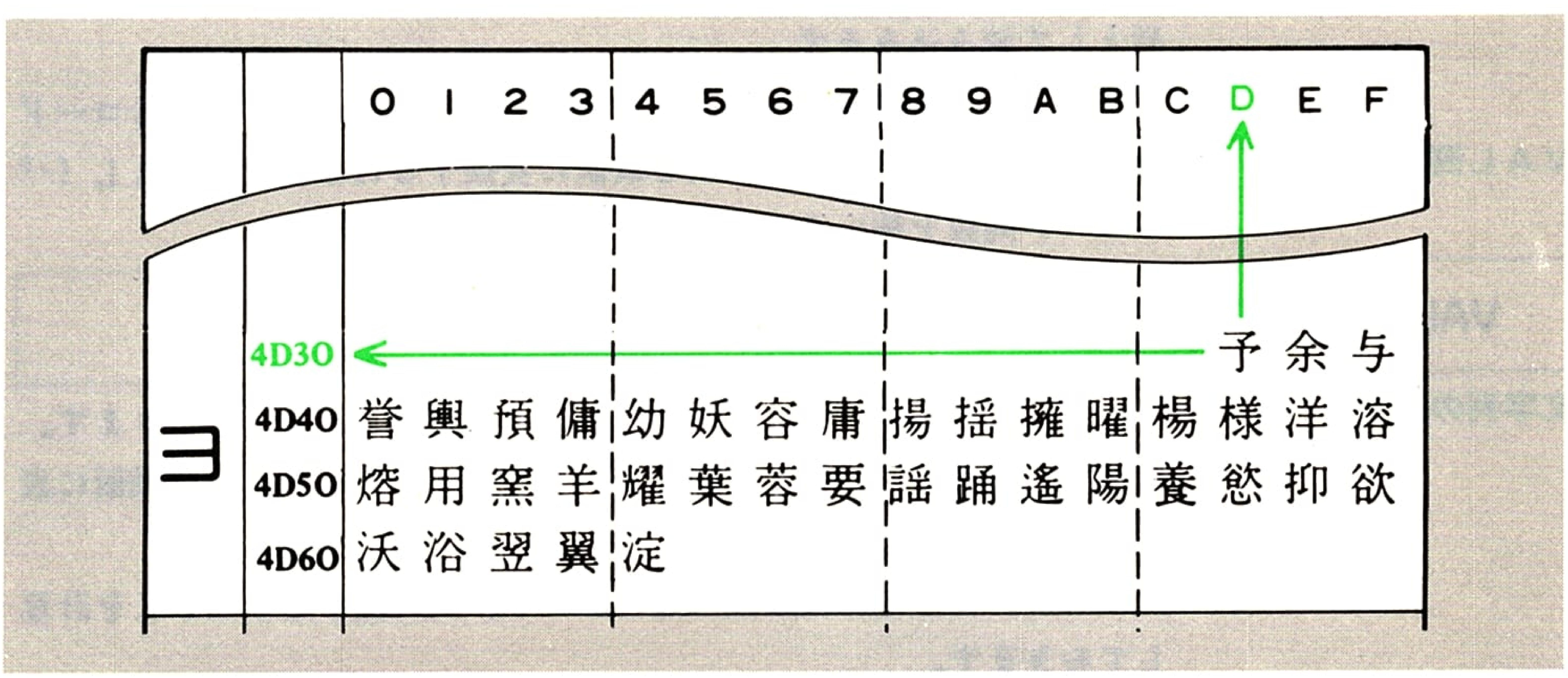
漢字をグラフィック画面に表示するには、次のPUT文を使います。　PUT文

```
PUT △ (X,Y),KANJI( 漢字JISコード),PSET,フォアグランドカラー,
バックグランウドカラー
```

このPUT文により、4桁の漢字JISコードで指定された漢字をドット（X，Y）に、縦16ドット、横16ドットの大きさで表示します。　漢字JISコード

漢字を漢字JISコードで指定するためには、『PC-8801 USER'.S MANUAL』の「19－7　漢字JISコード表」(19－17～19－29頁」を参照して、音読みで漢字を探す必要があります。　漢字JISコード表

たとえば、予算の予の漢字JISコードは、予の音読みはヨですから、まず、漢字JISコード表においてヨを探します。次に、予を探します。予が見つかったら、予から左横に見てコード番号（4D30）を探し、次に予から真上に見てコード番号（D）を探します。予の漢字JISコードは、4D3Dとなります。4D30Dとしてはいけません。漢字JISコードは、4桁です。



そこで、次のようにすれば、グラフィック画面のドット(159，0)に、予を白色で表示することができます。

```
PUT △(159,0),KANJI(4D3D),PSET,7,0
```

漢字タイトルをグラフィック画面に表示するには

漢字タイトル「予算実績差異分析表」をグラフィック画面に表示するためには、次のようにします。

まず、行番号3030のＤＡＴＡ文に、「予、算、実、績、差、異、分、析、表」に対応する９個の漢字ＪＩＳコードを用意しておきます。

次に、行番号3040のマルチステートメントにおいて、表示漢字数Ｎに９を代入し、ドット（Ｘ，Ｙ）のＸとＹを159と０に指定し、次の漢字との間のスペースの大きさＳを32とします。

そして、行番号3050のＧＯＳＵＢ文によって、行番号5000からはじまる漢字サブルーチンへ飛びます。漢字サブルーチンのプログラムは、次のとおりです。

```
5000 REM ***** SUB (KANJI) *****
5010 FOR I=1 TO N
5020 READ KJC$                        ←漢字ＪＩＳコードを文字列として読み込む
5030 KCODE=VAL("&H"+KJC$)             ←文字列の漢字ＪＩＳコードを数値に変換
5040 PUT (X,Y),KANJI(KCODE),PSET,7,0  ←漢字をグラフィック画面に表示
5050 X=X+S                            ←次の漢字のドットＸを計算
5060 NEXT I
5070 RETURN
```

まず、行番号5020のＲＥＡＤ文によって、漢字ＪＩＳコードを文字列として読み込みます。

ＶＡＬ関数

そして、行番号5030の代入文によって、文字列の漢字ＪＩＳコードを数値に変換します。文字列を数値に変換するには、次のＶＡＬ（バリュー）関数を使います。

```
VAL(文字列)
```

文字列の連結

なお、文字列は、＋（プラス）によって連結することができます。

次に、行番号5040のＰＵＴによって、漢字をグラフィック画面に表示します。

そして、行番号5050の代入文によって、次の漢字のドットＸを計算しておきます。

以上のことを行番号5010～5060のＦＯＲ～ＮＥＸＴ文によって、Ｎ回繰り返し、Ｎ個の漢字をグラフィック画面に出力します。

そして、行番号5070のＲＥＴＵＲＮ文によって、行番号3050のＧＯＳＵＢ文の次の行番号の文へもどります。

漢字見出しをグラフィック画面に表示するには

次に、同じようにして、漢字見出しをグラフィック画面に表示します。

ただし、「予算」、「実績」、「差異」の見出しは、タイトル「予算実績差異分析表」の最初の６個と同じ漢字ＪＩＳコードです。そこで、次のＲＥＳＴＯＲＥ文を使って、ＲＥＡＤ文で読むＤＡＴＡ文を指定します。

ＲＥＳＴＯＲＥ文

```
RESTORE △ 行番号
```

すなわち、行番号3070のＲＥＳＴＯＲＥ文によって、行番号3030のＤＡＴＡ文を指定します。

そして、行番号3080のＧＯＳＵＢ文によって、行番号3030のＤＡＴＡ文の最初の６個の漢字ＪＩＳコードの漢字を画面に表示します。

なお、行番号3030のＤＡＴＡ文の漢字ＪＩＳコードがまだ３個残っていますので、行番号3090～3110のＦＯＲ～ＮＥＸＴ文で読み込んでおきます。ただし、画面に表示はしません。

テキスト画面に数値や横線を表示するには

行番号3500～3600までの文によって、次のサブルーチンプログラムをコールし、テキスト画面に数値や横線を表示します。

```
6000 REM ***** SUB (###,###) *****
6010 COLOR 4
6020 PRINT
6030 PRINT SPC(12);
6040 PRINT USING " ###,###";B(I);
6050 PRINT USING " ###,###";A(I);
6060 PRINT USING " ###,###";V(I)
6070 RETURN
7000 REM ***** SUB (————) *****
7010 COLOR 2
7020 PRINT SPC(12);" ———— ———— ————"
7030 RETURN
8000 REM ***** SUB (════) *****
8010 COLOR 2
8020 PRINT SPC(12);" ════ ════ ════"
8030 RETURN
```

テキスト画面に数値を表示（6000～6070）

テキスト画面に１本線の横線を表示（7000～7030）

テキスト画面に２本線の横線を表示（8000～8030）

8・5 エンド部分とサブルーチンのプログラム

エンド部分のプログラムのリスト

予算実績差異分析プログラムのエンド部分のプログラムのリストは、次のとおりです。

```
4000 REM ***** END *****
4010 CLS 3
4020 COLOR 4
4030 PRINT M3$;" ｵﾜﾘ ";M3$
4040 PRINT
4050 PRINT " ｵﾂｶﾚｻﾏﾃﾞｼﾀ "
4060 PRINT
4070 PRINT " ｵﾜﾘﾃﾞｽ "
4080 GOSUB 9000
4090 CLS
4100 WIDTH 80,20
4110 CONSOLE 0,20,1,0
4120 END
```

サブルーチンプログラムのリスト

サブルーチンプログラムは、5つあります。そのうち4つは、すでに説明しました。残りの1つは、次のとおりです。

```
9000 REM ***** SUB (HIT ANYKEY) *****
9010 COLOR 1
9020 LOCATE 10,21
9030 PRINT M3$;" HIT ANYKEY ";M3$
9040 Z$=INKEY$
9050 IF Z$="" GOTO 9020
9060 BEEP
9070 RETURN
```

予算実績差異分析プログラムのリスト

ＬＩＳＴコマンドを使って、予算実績差異分析プログラムを画面に表示してみましょう。

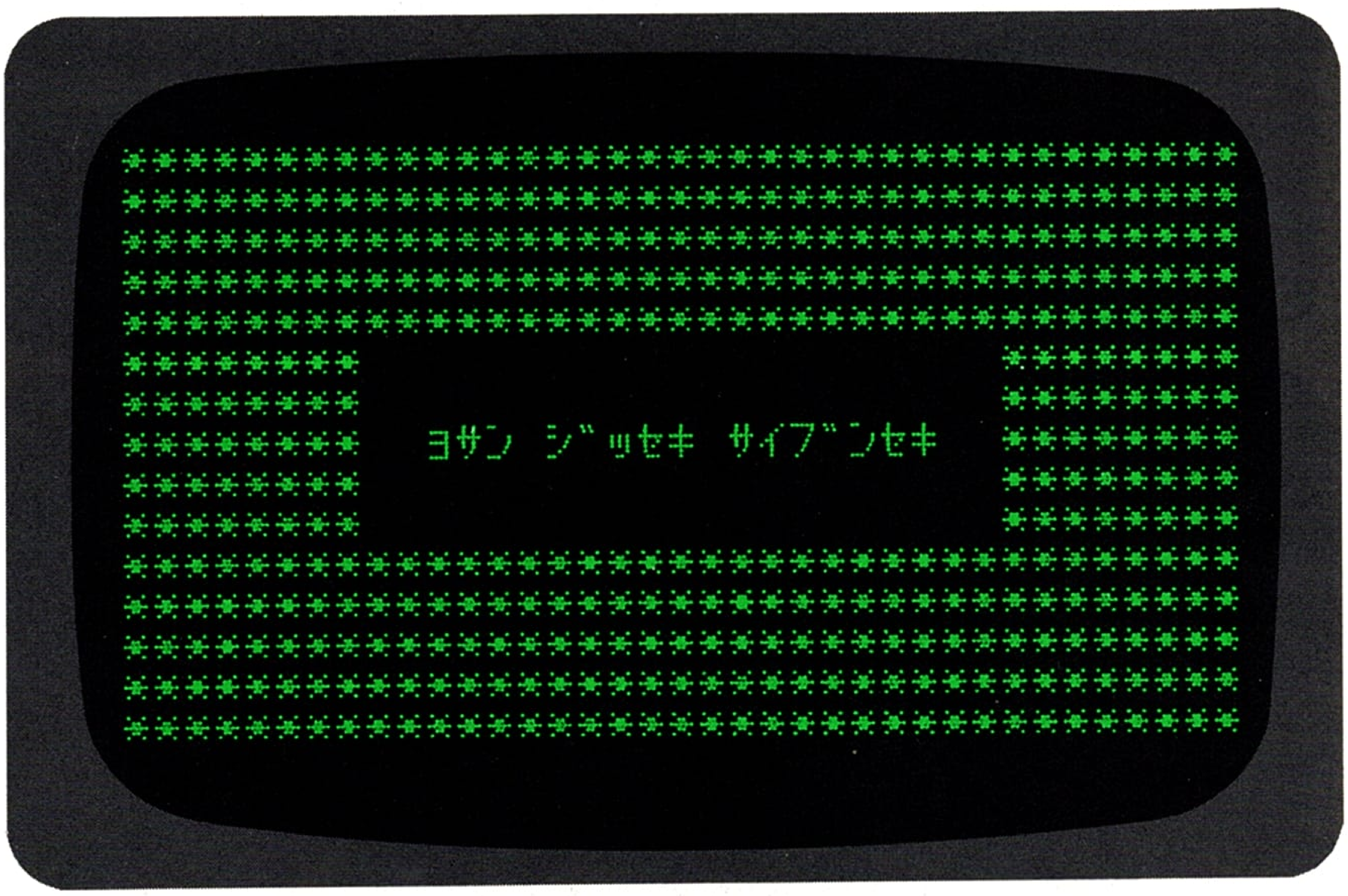
また、ＬＬＩＳＴコマンドを使って、予算実績差異分析プログラムのリストをプリンタに出力してみてください。

8・6　予算実績差異分析プログラムの実行

ＲＵＮコマンドを使って、予算実績差異分析プログラムを実行してみましょう。

初期画面

予算実績差異分析プログラムを実行すると、まず、ディスプレイに次のような初期画面が表示されます。



データのインプット

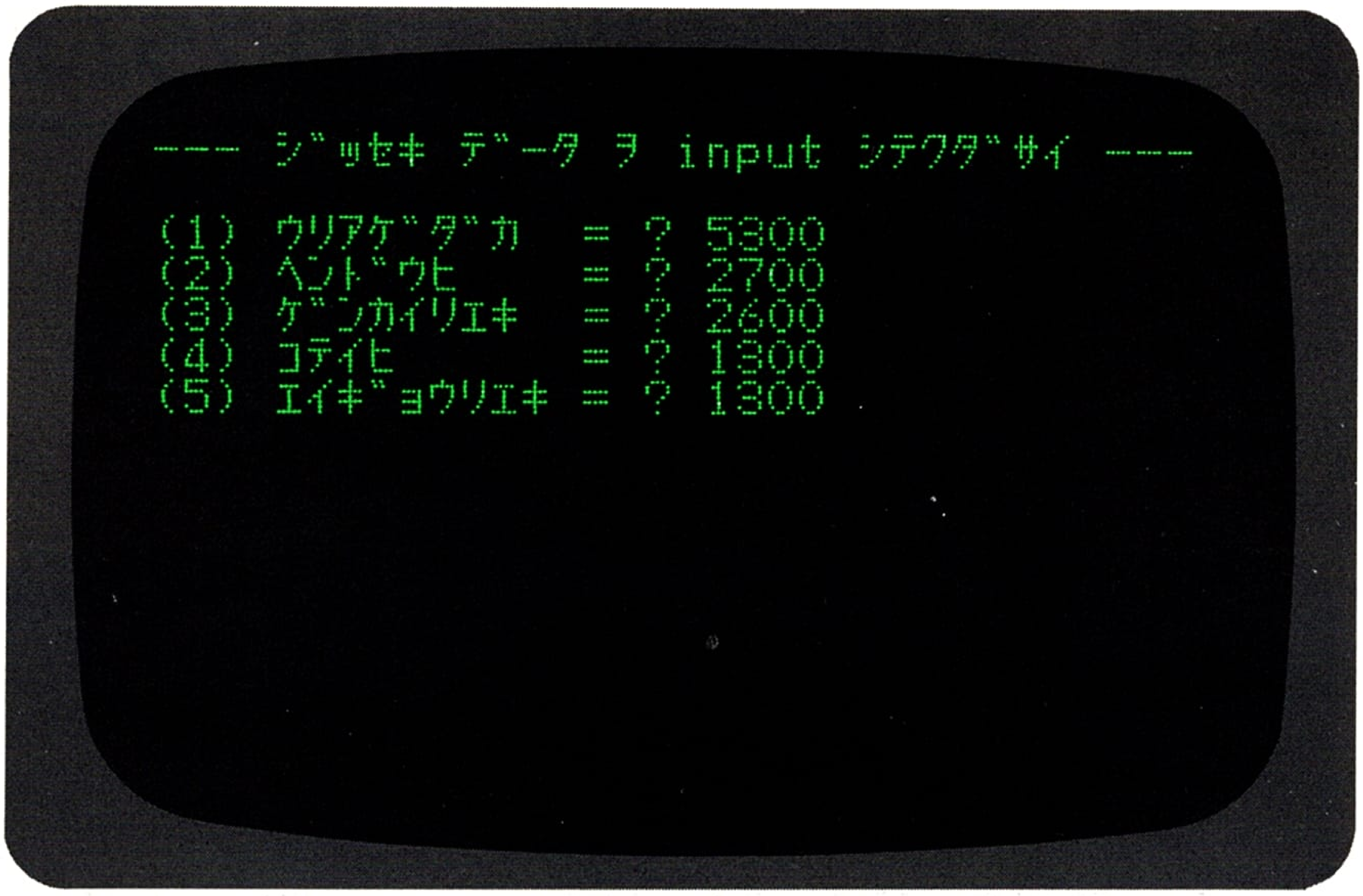
次に、画面のメッセージにしたがって、実績データをキーインします。いま、次ページ上の画面のように実績データをキーインしたとします。すなわち、売上高＝5,300万円、変動費＝2,700万円、限界利益＝2,600万円、固定費＝1,300万円、営業利益＝1,300万円です。

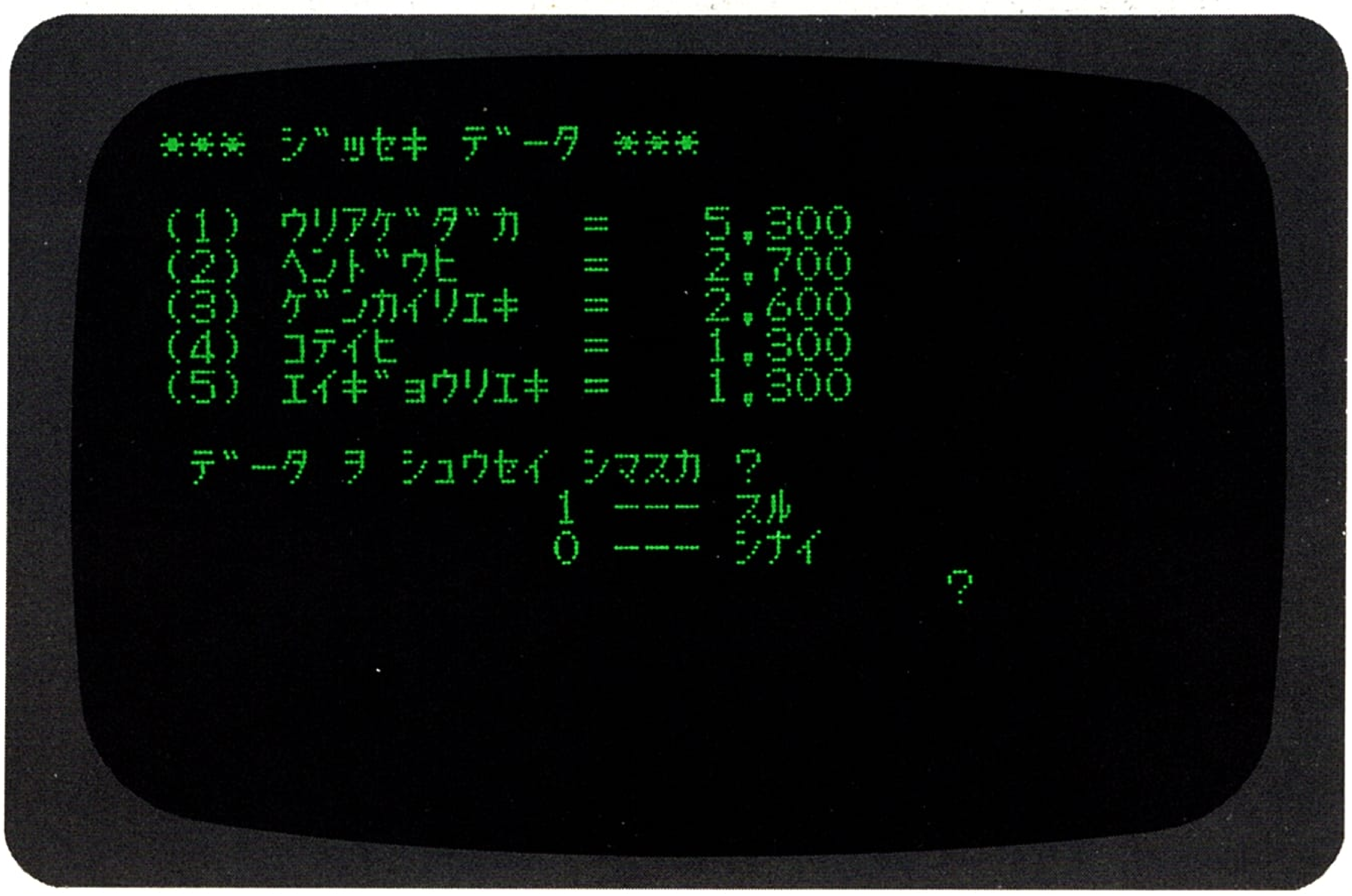
実績データの見やすい画面表示

ここで、⏎キーを押すと、次ページ下のように、実績データが画面に見やすく表示されます。

修正したいデータがあれば、[1] ⏎とし、修正したいデータの番号をキーインして、正しいデータをキーインします。

すべてのデータが正しくキーインできたら、[0] ⏎とします。画面に、予算実績差異分析表が表示されます。





予算実績差異分析表の表示

予算実績差異分析表は、次ページのとおりです。

ここで、どのキーでもいいから押せば、オワリの画面になります。

オワリ

オワリの画面が表示されたら、どれかキーを押せば、予算実績差異分析プログラムの実行が終わります。

予算実績差異分析表

	予算	実績	差異
売上高	5,000	5,300	300
変動費	2,500	2,700	200
限界利益	2,500	2,600	100
固定費	1,300	1,300	0
営業利益	1,200	1,300	100

--- HIT ANYKEY ---

8・7　予算実績差異分析プログラムのまとめ

予算実績差異分析プログラム

ここでの予算実績差異分析プログラムとは、売上高、変動費、限界利益、固定費、営業利益の予算データと実績データにもとづいて、それらの差異を計算し、予算実績差異分析表（タイトルおよび見出しを漢字で表示する）を作成するプログラムです。

予算実績差異分析表をプリンタに出力しよう

ＣＯＰＹ文あるいはＬＰＲＩＮＴ文やＬＰＲＩＮＴ　ＵＳＩＮＧ文を使って、予算実績差異分析表をプリンタに出力してみてください。

予算実績差異分析プログラムのセーブ

予算実績差異分析プログラムが完成したら、セーブ（ＳＡＶＥ）しておきましょう。

より良い予算実績差異分析プログラムにするには

ここで作成した予算実績差異分析表を、販売地域別に作成したり、製品別に作成したり、事業部別に作成したりすれば、より良い予算実績差異分析プログラムとなります。

また、ここでは、インプットデータが売上高、変動費、限界利益、固定費、営業利益の５項目でしたが、もっと詳細なインプットデータにすれば、くわしい予算実績差異分析表を作成することができます。

より良い予算実績差異分析プログラムを作成してみてください。

なお、ここでは、内部ファイルとしてＤＡＴＡ文を使いました。しかし、ＤＡＴＡ文による内部ファイルは、その大きさもかぎられますし、メンテナンスもたいへんな作業になります。そこで、フロッピィディスクユニットなど外部記憶装置を使いたいところです。しかし、この点については、プログラミング入門としての本書の範囲をこえます。次のステップというところです。

第9章

N-BASIC(DISK-BASIC)を使うには

9・1　N88-BASICとN-BASIC

2つのBASIC

2つのBASIC

PC-8801は、プログラミング言語として、N88-BASICとN-BASICの2つのBASICを持っており、必要に応じてきりかえて使います。

N-BASICは、PC-8001が持っているものとまったく同じです。N88-BASICは、N-BASICに多くの拡張機能をつけ加えたもので、PC-8801のメインプログラミング言語です。

N88-BASICとN-BASICの基本的なちがい

N88-BASICと
N-BASICの
基本的なちがい

N88-BASICとN-BASICの基本的なちがいは、次のとおりです。

(1) コマンド

N-BASICのCSAVE、CLOAD?、CLOADコマンドは、N88-BASICではなくなり、それぞれ、SAVE、LOAD?、LOADコマンドを使います。

また、DISK-BASICのMOUNT、REMOVE、FORMATコマンドは、N88DISK-BASICではなくなりました。

(2) 機能

N-BASICにおけるアトリビュート方式のセミグラフィック機能とキャラクタによって線を引く機能は、N88-BASICではなくなりました。

(3) 変数名

N-BASICでは最初の2文字までしか判別の対象になりませんが、N88-BASICでは40文字まで使用可能です。

(4) 実数値を整数値に変換する方法

N-BASICでは小数点以下を切り捨てて整数化しますが、N88-BASICでは小数点以下を四捨五入します。

(5) エラーメッセージの出力のしかた

N-BASIC(DISK-BASIC)やN88DISK-BASICでは、フルセンテンスのメッセージを出力し、N88-BASICでは省略形のメッセージを出力します。エラーメッセージについては、PC-8801 N88-BASIC N-BASIC REFERENCE MANUALを参照してください。

9・2 PC-8001用プログラムを使うには

ＰＣ-８００１用プログラムを使うには

ＰＣ-８００１用プログラムを使うには、N_{88}－ＢＡＳＩＣ（N_{88}ＤＩＳＫ－ＢＡＳＩＣ）からＮ－ＢＡＳＩＣ（ＤＩＳＫ－ＢＡＳＩＣ）にきりかえることが必要です。この場合、それまでにN_{88}－ＢＡＳＩＣ（N_{88}ＤＩＳＫ－ＢＡＳＩＣ）で入力されていたプログラムやデータは、すべて消えてしまいます。きりかえ前にセーブ（ＳＡＶＥ）してください。

Ｎ－ＢＡＳＩＣのスタート方法

Ｎ－ＢＡＳＩＣのスタート方法

ＮＥＷ　ＯＮ文

Ｎ－ＢＡＳＩＣのスタート方法は、次のとおりです。

❶　N_{88}－ＢＡＳＩＣをスタートさせます。

❷　次のＮＥＷ　ＯＮ文でＮ－ＢＡＳＩＣにきりかえます。

```
NEW△ ON △1 ⏎
```

❸　これでＮ－ＢＡＳＩＣモードになり、ＰＣ-８８０１本体前面の緑色のＬＥＤは消えます。そして、画面にＮ－ＢＡＳＩＣのメッセージが表示されます。

ＤＩＳＫ－ＢＡＳＩＣのスタート方法

ＤＩＳＫ－ＢＡＳＩＣのスタート方法

ＮＥＷ　ＯＮ文

ＤＩＳＫ－ＢＡＳＩＣのスタート方法は、次のとおりです。

❶　N_{88}ＤＩＳＫ－ＢＡＳＩＣをスタートさせます。

❷　ＤＩＳＫ－ＢＡＳＩＣ用システムディスクをミニフロッピィディスクユニットのドライブ１にセットします。

❸　次のＮＥＷ　ＯＮ文でＮ－ＢＡＳＩＣにきりかえます。

```
NEW△ ON △1 ⏎
```

❹　これでＮ－ＢＡＳＩＣモードになり、ＰＣ-８８０１本体前面の緑色のＬＥＤは消えます。そして、ミニフロッピィディスクが動き出し、ＤＩＳＫ－ＢＡＳＩＣがスタートし、画面にＤＩＳＫ－ＢＡＳＩＣのメッセージが表示されます。

N_{88}－ＢＡＳＩＣ（N_{88}ＤＩＳＫ－ＢＡＳＩＣ）にもどすには

リセットボタンを押すか、ＰＣ-８８０１本体の電源のＯＦＦ、ＯＮを使えば、N_{88}－ＢＡＳＩＣ（N_{88}ＤＩＳＫ－ＢＡＳＩＣ）にもどります。この場合、それまでにＮ－ＢＡＳＩＣ（ＤＩＳＫ－ＢＡＳＩＣ）で入力されていたプログラムやデータは、すべて消えてしまいます。注意してください。

9・3 次のステップへ

N_{88}－ＢＡＳＩＣの主な特徴

N_{88}－ＢＡＳＩＣの主な特徴

N_{88}－ＢＡＳＩＣの主な特徴は、次のとおりです。

❶ ＰＣ-８８０１が持っている機能のほとんどが使えます。

❷ 強力なグラフィック命令によって、多彩なグラフィックスが実現できます。

❸ テキスト画面とグラフィック画面の合成が簡単にできます。

❹ 多彩なカラー表示ができます。

❺ 大規模なプログラム作成が可能です。

❻ 日本語の文章を処理することができます。

❼ N_{88}ＤＩＳＫ－ＢＡＳＩＣにおいては、倍精度関数が使えます。

このように、N_{88}－ＢＡＳＩＣには多くの特徴があります。ＰＣ-８８０１を使いこなすには、これらの特徴をすべてカバーする必要があります。しかし、この点については、ＰＣ-８８０１のプログラミング入門としての本書の範囲をこえます。

おわりに

これで、ＰＣ-８８０１のプログラミング入門は終わります。どうもお疲れさまでした。いかがですか。ＰＣ-８８０１に慣れましたか。しかし、ＰＣ-８８０１を使いこなすのはこれからです。

ＰＣ-８８０１は、事務処理、技術計算、教育、端末、計測制御など、幅広い利用が可能です。あなたの興味は、どこにありますか。次のような点に興味はありませんか。

❶ ディスクモードのデータファイル（シーケンシャルファイルとランダムアクセスファイル）の利用

❷ 多彩なビジネスグラフィックスの利用

❸ 日本語情報処理

ＰＣ-８８０１を使いこなせば、これらの点はすべて実現可能です。

次のステップへ挑戦

ぜひ、次のステップへ挑戦してください。期待しています。

索引（和文：英文）

著者略歴

野々山隆幸（ののやま　たかゆき）

1947年２月27日愛知県生まれ。
1971年３月一橋大学大学院商学研究科修士課程修了。
横浜市立大学商学部助手となり、専任講師をへて、
1975年４月に助教授となり、現在に至る。
現住所＝横浜市緑区長津田町2469
　　　　　東向地団地１－122（〒227）

PC-8801
プログラミング入門

著　者　野々山隆幸
発行者　田村　正隆

発行所　株式会社　ナツメ社
　　　　東京都千代田区神田神保町１—52（〒101）
　　　　電話　03(291)1257（代表）
　　　　振替　東京３—58661
印　刷　ラン印刷社
製　本　文章堂製本

N88－BASIC (N88 DISK－BASIC) の主なコマンドとステートメント(文)

（△はスペースを１つあける ことを意味します）

1 AUTO

▶AUTO△行番号 ⏎
⏎キーを押すたびに、指定した行番号からはじまって、10きざみに増加する行番号を自動的に発生します。行番号を省略すると、10が選ばれます。STOP キーを押すと、行番号の自動発生がストップします。

2 BEEP

▶BEEP ⏎
内蔵スピーカを鳴らす（少しの間）。
▶BEEP△1 ⏎
内蔵スピーカを鳴らす（鳴りっぱなし）。
▶BEEP△0 ⏎
内蔵スピーカの音を止める。

3 CLS

▶CLS⏎またはCLS△1⏎
テキスト画面のみをクリア
▶CLS△2 ⏎
グラフィック画面のビューポート内のみをクリア
▶CLS△ ⏎
テキスト画面とグラフィック画面の両方をクリア

4 COLOR

▶COLOR△カラーコード ⏎
カラーモードのテキスト画面の文字の色を次のカラーコードによって指定します。

0――黒	4――緑
1――青	5――水色
2――赤	6――黄色
3――紫	7――白

5 CONSOLE

テキスト画面のモードを設定します。
▶CONSOLE△，，，0⏎
白黒モードにします。
▶CONSOLE△，，，1⏎
カラーモードにします。
▶CONSOLE△，，0，0⏎
ファンクションキーの表示を消し、白黒モードにします。
▶CONSOLE△，，0，1⏎
ファンクションキーの表示を消し、カラーモードにします。
▶CONSOLE△，，1，0⏎
ファンクションキーの表示をし、白黒モードにします。
▶CONSOLE△，，1，1⏎
ファンクションキーの表示をし、カラーモードにします。

6 COPY

プリンタに画面コピー（スクリーンコピー）をとります。
▶COPY ⏎
テキスト画面とグラフィック画面の両方をコピー
▶COPY△1 ⏎
テキスト画面のみコピー
▶COPY△2 ⏎
グラフィック画面のみコピー
▶COPY△3 ⏎
テキスト画面とグラフィック画面の両方をコピー
▶COPY△4 ⏎
グラフィック画面のみコピー（漢字出力用）
▶COPY△5 ⏎
テキスト画面とグラフィック画面の両方をコピー（漢字出力用）

7 DATA

▶DATA△定数，定数，…⏎
READ文で読み込まれる数値を用意しておきます。１つのDATA文には、１行（255文字）に入るだけのデータをセットできます。
（→READを参照）

8 DIM

▶DIM△配列変数名（添字の最大値），配列変数名（添字の最大値），… ⏎
配列変数名と添字の最大値を指定し、配列の宣言をします。

9 END

▶END ⏎
プログラムの実行を終了します。

10 FILES

▶FILES ⏎ またはFILES△1 ⏎
ドライブ１のディスクに入っているファイルの名前、種類、大きさをディスプレイに表示します。
▶FILES△2 ⏎
ドライブ２のディスクです。

11 FOR…TO～NEXT

▶FOR△変数名＝初期値△TO△最終値 ⏎
～
NEXT△変数名 ⏎
変数の値が初期値から１きざみで最終値になるまで、FOR文からNEXT文の間の一連の文を繰り返し実行します。

12 GOSUB～RETURN

▶GOSUB△行番号 ⏎
～
RETURN ⏎
指定した行番号からはじまるサブルーチンプログラムをGOSUB文によってコールし、RETURN文によってGOSUB文の次の文へリターンします。

13 GOTOまたはGO△TO

▶GOTO△行番号 ⏎
指定した行番号の行へ飛びます。

14 IF…GOTO

▶IF△変数名＝数値△GOTO△行番号 ⏎
変数の値が指定した数値に等しければGOTO文以降を実行し、そうでなければ次の行を実行します。

15 IF…THEN

▶IF△変数名＝数値△THEN△ステートメント（文）⏎
変数の値が指定した数値に等しければTHEN文以降を実行し、そうでなければ次の行を実行します。

16 INPUT

▶INPUT△〝メッセージ〞；変数名 ⏎
画面にデータの入力をうながすメッセージと？（クエッションマーク）が表示されます。
データをキーインし、⏎キーを押すと、データが変数に代入されます。

17 KEY

▶KEY△キー番号，〝定義内容〞⏎
ファンクションキーの内容を定義します。キー番号は、１から10までです。定義内容は、最大15文字までです。

18 KEY△LIST

▶KEY△LIST ⏎
ファンクションキーのすべての内容を画面に表示します。

19 LET

数値、式の値、文字列を変数に代入します。LETは省略して、次のようにキーインします。
▶変数名＝数値 ⏎
▶変数名＝式 ⏎
▶文字型変数名＝〝文字列〞⏎

20 LFILES

ディスクに入っているファイルの名前、種類、大きさをプリンタに出力します。
▶LFILES ⏎
▶LFILES△1 ⏎
▶LFILES△2 ⏎
（→FILESを参照）

21 LINE

▶LINE△（X_1，Y_1）－（X_2，Y_2），色番号 ⏎
グラフィック画面において、ドット（X_1、Y_1）とドット（X_2、Y_2）を結ぶ直線を指定した色で引きます。

N88－BASIC (N88 DISK-BASIC) の主なコマンドとステートメント(文)

(△はスペースを１つあけることを意味します)

22 LIST

▶ＬＩＳＴ⏎
　プログラムの全部をディスプレイに表示します。

▶ＬＩＳＴ△行番号⏎
　指定した行番号の行のみを表示します。

▶ＬＩＳＴ△行番号－⏎
　指定した行番号からはじまってそれより大きい行番号の行をすべて表示します。

▶ＬＩＳＴ△－行番号⏎
　プログラムの最初の行番号から指定した行番号までの行を表示します。

▶ＬＩＳＴ△行番号－行番号⏎
　指定した行番号から行番号までの範囲のすべての行を表示します。

23 LLIST

　プログラムの全部または一部をプリンタに出力します。

▶ＬＬＩＳＴ⏎

▶ＬＬＩＳＴ△行番号⏎

▶ＬＬＩＳＴ△行番号－⏎

▶ＬＬＩＳＴ△－行番号⏎

▶ＬＬＩＳＴ△行番号－行番号⏎

(→ＬＩＳＴを参照)

24 LOAD

▶ＬＯＡＤ△〝プログラム名〟⏎
　ディスクのプログラムをＰＣ－８８０１のメモリにロードします。

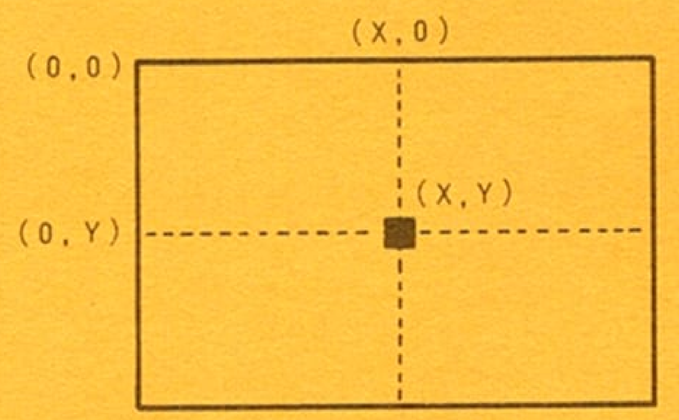
▶ＬＯＡＤ△〝ＣＡＳ：プログラム名〟⏎
　カセットテープのプログラムをメモリにロードします。

25 LOAD?

▶ＬＯＡＤ?△〝ＣＡＳ：プログラム名〟⏎
　カセットテープに正しくプログラムがセーブできたかどうかを確認（ベリファイ）します。

26 LOCATE

▶ＬＯＣＡＴＥ△Ｘ，Ｙ⏎
　テキスト画面のカーソルの位置をＸ桁Ｙ行に指定します。



27 LPRINT

　プリンタに情報を出力します。

▶ＬＰＲＩＮＴ△式⏎

▶ＬＰＲＩＮＴ△〝メッセージ〟；変数名⏎

▶ＬＰＲＩＮＴ⏎

(→ＰＲＩＮＴを参照)

28 LPRINT USING

▶ＬＰＲＩＮＴ△ＵＳＩＮＧ△〝＃，＃＃＃，＃＃＃〟；変数名⏎
　変数の値を右づめで３桁ごとにコンマ（，）で区切って、プリンタに出力します。

29 MOTOR

▶ＭＯＴＯＲ⏎
　カセットテープレコーダのモータがＯＮの状態ならＯＦＦに、ＯＦＦの状態ならＯＮにします。

▶ＭＯＴＯＲ△０⏎
　モータをＯＦＦにします。

▶ＭＯＴＯＲ△１⏎
　モータをＯＮにします。

30 NEW

▶ＮＥＷ⏎
　ＰＣ－８８０１のメモリの中にあるプログラムを消去します。

31 NEW ON

▶ＮＥＷ△ＯＮ△１⏎
　N88－ＢＡＳＩＣからＮ－ＢＡＳＩＣにきりかえます。

32 ON…GOTO

▶ＯＮ△変数名△ＧＯＴＯ△行番号，行番号，…
　変数の値が１なら１番目の行番号の行に飛び、変数の値が２なら２番目の行番号の行へ飛びます。

33 PRINT

▶ＰＲＩＮＴ△式⏎
　ディスプレイに式の値を出力します。

▶ＰＲＩＮＴ△〝メッセージ〟；変数名⏎
　ディスプレイにメッセージと変数の値を出力します。

▶ＰＲＩＮＴ⏎
　ディスプレイに１行空白行を出力します。
　なお、ＰＲＩＮＴのかわりに？（クエッションマーク）を使うことができます。

34 PRINT USING

▶ＰＲＩＮＴ△ＵＳＩＮＧ△〝＃，＃＃＃，＃＃＃〟；変数名⏎
　変数の値を右づめで３桁ごとにコンマ（，）で区切って、ディスプレイに表示します。

35 PUT

▶ＰＵＴ△（Ｘ，Ｙ），ＫＡＮＪＩ（漢字コード），ＰＳＥＴ，７，０⏎
　グラフィック画面（Ｘ，Ｙ）に、漢字コードで指定した漢字を白色で表示します。

36 READ

▶ＲＥＡＤ△変数名⏎
　行番号の小さいＤＡＴＡ文から順に数値を変数に１対１に対応させて代入します。特定の行番号のＤＡＴＡ文を読むには、ＲＥＳＴＯＲＥ文を使います。

▶ＲＥＳＴＯＲＥ△行番号⏎
(→ＤＡＴＡを参照)

37 REM

▶ＲＥＭ△注釈⏎
　プログラムに注釈(コメント)を入れます。
　なお、ＲＥＭのかわりに'（アポストロフィ）を使うことができます。

38 RENUM

▶ＲＥＮＵＭ⏎
　プログラムの行番号を10きざみにつけかえます。

▶ＲＥＮＵＭ△新行番号，旧行番号⏎
　旧行番号を新行番号にかえ、以後10きざみにつけかえます。

39 RUN

▶ＲＵＮ⏎
　ＰＣ－８８０１のメモリにあるプログラムを実行します。

▶ＲＵＮ△〝プログラム名〟⏎
　ディスクからプログラムをメモリにロードし、そのプログラムを実行します。

40 SAVE

▶ＳＡＶＥ△〝プログラム名〟⏎
　ＰＣ－８８０１のメモリのプログラムをディスクにセーブします。

▶ＳＡＶＥ△〝ＣＡＳ：プログラム名〟⏎
　メモリのプログラムをカセットテープにセーブします。

41 SCREEN

　グラフィック画面のモードを設定します。

▶ＳＣＲＥＥＮ△０⏎
　カラーモードにします。

▶ＳＣＲＥＥＮ△１⏎
　白黒モードにします。

▶ＳＣＲＥＥＮ△２⏎
　高分解能白黒モードにします。

42 WIDTH

▶ＷＩＤＴＨ　40，20⏎
　画面を40桁×20行にします。

▶ＷＩＤＴＨ　40，25⏎
　画面を40桁×25行にします。

▶ＷＩＤＴＨ　80，20⏎
　画面を80桁×20行にします。

▶ＷＩＤＴＨ　80，25⏎
　画面を80桁×25行にします。

PC-8801
プログラミング入門

発行——昭和57年6月30日
著者——野々山隆幸
発行者——田村正隆
発行所——株式会社ナツメ社
郵便番号101
東京都千代田区神田神保町1-52
電話〈03〉291-1257
振替　東京3-58661
〈落丁・乱丁本はお取り替えします〉

定価——1400円

PC-8801
プログラミング入門

ナツメ社
定価＝1400円
2054-29106-5619

［目次より］